Aquaria: A Web-Based AI Platform Utilizing Convolutional Neural Networks and YOLOv8 for Precision Fishing, Aquaculture, and Marine Sustainability in the Philippines

A Research Study Presented to the

College of Technology

LYCEUM OF THE PHILIPPINES UNIVERSITY, MANILA

In Partial Fulfillment

Of the Requirements for the Degree of

Bachelor of Science in Computer Science

By:

Lanz Vincent T. Vencer

June 2023

Abstract

Aquaria is a pioneering research project offering a comprehensive suite of eight AI-powered services tailored to address sustainability challenges within the fisheries and aquaculture sectors in the Philippines. Utilizing innovative artificial intelligence methodologies such as the Cut Paste and Learn (CPL) technique for image synthesis and You Only Look Once (YOLO) for object detection, Aquaria provides efficient and effective solutions for diverse applications such as fish species identification, marine debris detection, and market marine animals recognition. The underlying models for these services have been rigorously trained using comprehensive datasets, and tested using established metrics such as Precision, Recall, and Mean Average Precision (mAP), culminating in high-performance, reliable tools. Furthermore, the use of Gradient-weighted Class Activation Mapping (Grad-CAM) not only improved model interpretability, but also enhanced model debugging, leading to refined and improved performance. The Aquaria services are seamlessly integrated and hosted on Streamlit, an open-source Python library, providing global accessibility and user-friendly interaction with the models. Also, the deployment of the services on Streamlit Cloud ensures continuous uptime and easy sharing, promoting global reach and impact. Through a thoughtful marriage of technology and marine science, Aquaria illustrates the transformative potential of artificial intelligence in promoting sustainable practices, providing crucial support for data-driven decision-making in marine management and conservation.

Lyceum of the Philippines University

Muralla cor Real St., Intramuros Manila

College of Technology

Adviser's Recommendation Sheet

The Thesis Project entitled "Aquaria: A Web-Based AI Platform Utilizing Convolutional Neural Networks and YOLOv8 for Precision Fishing, Aquaculture, and Marine Sustainability in the Philippines" developed and submitted by Lanz Vincent T. Vencer in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and endorsed for final defense as bounded by the stipulated objectives, scopes and delimitations.

Dr. Erlito M. Albina
Thesis Adviser

College of Technology

APPROVAL SHEET

This thesis entitled "**AQUARIA: A WEB-BASED AI PLATFORM UTILIZING CONVOLUTIONAL NEURAL NETWORKS AND YOLOV8 FOR PRECISION FISHING, AQUACULTURE, AND MARINE SUSTAINABILITY IN THE PHILIPPINES**" prepared and submitted by **LANZ VINCENT T. VENCER** has been reviewed and recommended as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Dr. Erlito M. Albina
Research Adviser 2

Dr. Arlene R. Caballero
Research Adviser 1

Accepted and approved by the panel of examiners following a successful Oral Defense and Presentation on June 18, 2023, with a grade of _________.

Mr. Gary R. Soriano
Panelist

Ms. Porsche L. Amo
Panel Chair

Dr. Arlene R. Caballero
Panelist

Accepted and approved in partial fulfillment of the requirements for the degree Bachelor of Science in Computer Science.

DR. ARLENE R. CABALLERO
Dean

Date: _____________

Acknowledgment

I would like to express my deepest gratitude to my Mother and Father for their utmost and unconditional support throughout the course of my studies. Without their love and encouragement, I would not have been able to complete my thesis.

I would also like to extend my sincere thanks to Rehmar Secocher for helping me by providing the materials for building the controlled environment, without which this research would not have been possible.

I am deeply indebted to my thesis advisers, Dr. Arlene Caballero and Dr. Erlito Albina, for their invaluable guidance and support throughout the research process. Their insights and expertise have been invaluable in shaping the direction of my work.

I am also grateful to the Kaggle Data Science & Machine Learning community for their suggestions and feedback, which have greatly improved the quality of my work.

Finally, I would like to thank all the other people who extended their support to provide the fishes and made this research possible. Thank you all.

Dedication

To my family, who have been with me every step of the way. To my mother and father, thank you for your utmost and unconditional support throughout my academic journey. Your love and encouragement have been my guiding light, and I am forever grateful for all that you have done for me,

To the fishermen and people making a livelihood in fisheries, who tirelessly work to provide for their families and communities, this thesis is dedicated to you. Your strength and resilience in the face of adversity inspired me to continue fighting for a better future. May this work bring about positive change and help uplift your lives, despite the challenges of poverty and exploitation that you face daily, and

To the makers and builders everywhere, to those risking failure one last time: This thesis is dedicated to all of you, the dreamers and doers who refuse to settle for mediocrity. You are the driving force behind the progress and innovation of our world. Without your tireless work and determination, the impossible becomes just that. But you continue to push the boundaries, challenge the status quo, and strive for greatness. Your passion and perseverance are a constant reminder to never give up on my dreams and to always aim for the stars. Thank you for your inspiration and for being a constant source of motivation in my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AIaaS	Artificial Intelligence as a Service
API	Application Programming Interface
ANN	Artificial Neural Network
AUV	Autonomous Underwater Vehicle
BP	Backpropagation
BR	Bayesian Regularization
CAD	Computer-Aided Design
CLI	Command Line Interface
COCO	Common Object in Context
CPL	Cut, Paste, Learn
CRM	Coastal Resource Management
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DOD	Definition of Done
FAIR	Facebook AI Research
FC	Fully Connected
GDP	Gross Domestic Product
GPU	Graphics Processing Unit
IoT	Internet of Things
IOU	Intersection Over Union
JSON	JavaScript Object Notation
LM	Levenberg-Marquardt
ML	Machine Learning
mAP	Mean Average Precision
MT	Metric Tons

PNG	Portable Network Graphic
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RPN	Region Proposal Network
SaaS	Software as a Service
SVM	Support Vector Machines
Tanh	Hyperbolic Tangent
UI	User Interface
UX	User Experience
YOLO	You Only Look Once

Chapter 1

PROBLEM AND ITS BACKGROUND

1.1 INTRODUCTION

The Philippines has been a major contributor to global fisheries for many years. The country was the fourth-largest producer of seaweed in the world and ranked 13th in terms of fish output in 2018, with a total production of roughly 4.36 million metric tons (MT) from three sectors, including aquaculture, municipal, and commercial fisheries that brought in US$1.6 billion (Food and Agriculture Organization of the United Nations [FAO], 2020; TahiLuddiN & Terzi, 2021). Furthermore, the contribution of the fishing sector to the country's Gross Domestic Product (GDP) in the same year was 1.2% and 1.3% at current and constant prices, respectively. In 2019, there were approximately 1.99 million fishers and 0.35 million fish growers in the nation, making fishing one of the main sources of income. Each Filipino consumed fish and fisheries products at an average of 34.27 kg/year in 2019, which included 23.36 kg of fresh fish, 2.85 kg of dried fish, 4.97 kg of processed fish, and 3.10 kg of crustaceans and mollusks (South East Asian Fisheries Development Center [SEAFDEC], n.d.). However, despite the growing and consistent output yield procured by the industry to help the country's economic growth, there are also negative tradeoffs incurred to the marine ecosystem, especially in the utilization of traditional ways and procedures of fishing and fish farming (Belton et al., 2020; Rana et al., 2022). As an example, the use of fishing gears and nets was shown to have a significant negative impact on marine and coastal fauna, entangling thousands of invertebrates as well as dugongs, crocodiles, sawfish, hammerhead sharks, and sea snakes (Gilman et al., 2021; Gunn et al., 2010; Hardesty et al., 2021). These external materials could cumulatively become harmful marine garbage and debris that could be a potential detriment to the marine ecosystem. The illegal practice of dynamite fishing also exacerbates the degradation because it causes negative impacts on the marine habitat, displacing many fish species and their sources of food (Breckwoldt et al., 2019; Hampton-Smith et al., 2021; Katikiro & Mahenge, 2016). Therefore, the need for a new and innovative methodology for fishing is significant and urgent to mitigate the irrevocable causes prevalent when using traditional methods. This will benefit the marine ecosystem and the people who make a living through fishing operations.

Precision fishing is the practice of maximizing fishing operations and management via the use of cutting-edge instruments and technologies (Costello et al., 2016). By using these high-tech methods of fishing, we can only take from the oceans what we need to survive, minimizing our detrimental effects on its healthy state. According to recent estimates, the ocean could sustainably produce six times as much food as it does now with better management. We can take care of our marine ecosystem and consume resources without hampering the existing balance through precision fishing which can aid in these endeavors (SafetyNet Technologies, 2021). With the advancements in Computer Science and Artificial Intelligence (AI), we could leverage the capabilities of deep learning algorithms and our everyday computing devices to develop a web service that could help identify efficient and optimal fishing strategies with minimal risk to the marine ecosystem. Various studies utilize machine learning and deep learning to aid in equipping the fishers with the right information to perform their everyday tasks more efficiently and effectively by providing data about marine behavior (Asche & Smith, 2018; Iqbal et al., 2022; D. Li & Li, 2020; Zhao et al., 2021). We could apply these studies to improve and further develop our existing technology-based solutions to help modernize our fishing strategies to maximize profit generation and food production without exploiting our marine resources.

The necessity for developing innovative solutions that can build on each other is growing in significance within the fisheries and aquaculture sector of the world (Asche & Smith, 2018). Having accurate harvest estimates is also significant for sustainable supply and demand management as recreational fishing expands and continually develops (Taylor et al., 2011). Furthermore, the achievement of the economic, social, and environmental goals in fisheries is facilitated by the sustainable production of fish and seafood products which is dependent on the strategies that are applied to reach these goals (Tolentino-Zondervan & Zondervan, 2022). Therefore, the application of modern ideas to improve the fishing sector should be a priority, especially in keeping up with global trends and growth. This study focuses on helping the people involved or making a living in fisheries and aquatic resources utilize artificial intelligence applications in their daily fishing operations. The researchers will try to provide AI-based solutions to help them practice precision fishing and efficiently fish for food while mitigating the risk of harming the marine ecosystem. The developed web application would also help the marine ecosystem recover from the detrimental effects caused by traditional methods of fishing and revert to its healthy structure.

1.2 OBJECTIVES OF THE STUDY

1.2.1 General Objective:

This study aims to provide an AI-based solution deployed in a web application to help fisheries and aquaculture beneficiaries practice precision fishing and improve fishing efficiency by curating datasets available from related studies and synthetically generated image data. These datasets will be used as training, validation, and testing data to develop Computer Vision models that will eventually be placed into production.

1.2.2 Specific Objectives:

1. To curate a large-scale dataset and generate synthetic images for different use-cases from online repositories, previous relevant studies, and data available on the internet using data engineering and data mining methods;
2. To develop a Convolutional Neural Network (CNN) algorithm for binary and multi-class classification and identify the best-performing CNN model through different metrics using the validation and test datasets;
3. To utilize state-of-the-art Computer Vision algorithms like You Only Look Once (YOLO) for Object Detection and Object Tracking with the SORT method;
4. To develop an AI Web Service hosted on Streamlit that could be accessed by the target users; enabling them to upload images and get results based on the prediction of the embedded model trained using the datasets;
5. To evaluate the system based on ISO 25010 in terms of the following:
    a. Functional Suitability
    b. Performance Efficiency
    c. Compatibility
    d. Usability
    e. Reliability
    f. Security
    g. Maintainability
    h. Portability

1.3. SIGNIFICANCE OF THE STUDY

The Philippines, being an archipelago surrounded by different bodies of water, has a lot of potential for maximizing economic growth and food production in the aquaculture and fisheries industry. Developing innovative and efficient tools and methods for fishing through the solutions outlined in this study will help in taking advantage of the resources present within the water bodies surrounding the country while also considering the state of the marine ecosystem to avoid overexploitation. Furthermore, this study can also serve as a framework to provide an innovative solution using Computer Vision to aid the target beneficiaries in utilizing precision fishing through the features present in the web application. With this application being placed into production, we could narrow the gap between the technical knowledge of utilizing artificial intelligence and democratize it to more people without the needed prerequisites to utilize such technology. Specifically, this study can be highly significant for the following:

- ***Fishermen and Fisherfolks***

This study hopes to develop an AI as a Service to help fishermen efficiently and effectively optimize their everyday tasks of catching fish using their everyday computing devices on demand. They could utilize the web service to remotely identify and locate the presence of different marine animals by uploading the images captured by an Autonomous Underwater Vehicle (AUV), statically placed monitoring or image-capturing device, or human divers submerged underwater.

- ***Marine Biologists and Fish Farm Workers***

The proposed web application can serve as an automation pipeline for monitoring the state of the marine ecosystem and the organisms living within an area of study or production. Marine Biologists and Fish Farm Workers could utilize the computer vision features to generate reports once a certain fish stop moving or when the presence of a predatory species was introduced within the ecosystem. Furthermore, they could also extend the feature to different domains of marine biology to cater to their respective specializations.

- ***Marine Sustainability and Safety Advocates***

The study could empower marine sustainability and safety advocates in their cause of protecting the ocean and avoiding exploitation of the marine resources. The proposed solution could provide fishermen the tools to practice precision fishing using computer vision algorithms to determine optimal locations of fish within the monitored zones and prevent rampant misuse of fishing nets and unattended gear that could cause long-term effects on marine wildlife.

- ***Data Engineers***

The proposed Synthetic Image Generation algorithm would help data engineers develop pipelines to resolve problems involving the lack of training data. Furthermore, they could use this approach to create CNN models for domain-specific tasks. This will also be beneficial in saving time and cost for collecting and procuring datasets to be used for their own needs.

- ***Future Researchers***

This study and its proposed solutions could help future researchers utilize the methodologies as a basis for extending the capabilities of the web application in other types of ecosystems. The approach of creating synthetic images as training datasets could be utilized in other domains, particularly within medicine, biology, epidemiology, and automation. Furthermore, this could also serve as a basis for the use of synthetic images as an effective approach for computer vision tasks such as image recognition, classification, and segmentation. The compatibility of the proposed algorithms could also be further migrated to a more sophisticated Computer Vision model that does not exist yet as of the publication of this paper.

1.4. SCOPE AND DELIMITATIONS

The primary focus of this study is to develop a web-based AI service, "Aquaria", aimed at pioneering various computer vision innovations in the realm of precision fishing, aquaculture, and marine operations in the Philippines. The development process of the project will be conducted in an iterative format, where each iteration delivers a functional component of the overall system. The major iterations include:

1. The first iteration involves creating an AI service for precision litter detection and classification using the YOLO v8 object detection model. This service will help identify and classify various types of marine litter, aiding in efforts to monitor and reduce pollution in the aquatic environment.

2. The second iteration will develop an AI service for marine animal image classification using transfer learning. This service will aid in the identification and classification of various marine species, supporting biodiversity studies and ecological monitoring.

3. The third iteration entails the development of a unique AI service that utilizes synthetic images for market marine animal classification. This service will implement the Cut, Paste, and Learn algorithm to generate synthetic image data for model training.

Python will be the main programming language for model development and deployment, leveraging modern frameworks such as Pytorch, TensorFlow, and Keras. The web application will be built and deployed using the Streamlit framework due to its seamless integration with Python and its ability to quickly prototype and deploy data-intensive applications.

The following are the limitations of the study:

1. The performance of the AI models developed in this study will be heavily dependent on the quality and quantity of the training data. The use of synthetic images for training, while beneficial in scenarios with limited real-world data, may not completely reflect real-world complexities.

2. While the services developed in this study are intended for use in precision fishing, aquaculture, and marine operations, their effectiveness in real-world applications will need to be evaluated in the context of these operations.

3. The web-based services developed in this study are designed to perform specific tasks within computer vision. They may not be directly applicable to other domains without modifications or further development.

4. The application will be developed for web-based deployment. Therefore, its performance might be affected by network conditions, and it may not offer the same level of accessibility or user experience as a native application on certain devices.

1.5 DEFINITION OF TERMS

1.5.1 Technical Definition of Terms

Aquaculture – Aquaculture refers to all activities related to the farming, raising, and cultivation of fish and other aquatic species in fresh, brackish, and marine water environments (*The Philippine Fisheries Code of 1998.*, n.d.).

Aquafarms – facilities that are utilized for the purpose of culturing or propagating aquatic species, such as fish, mollusks, crustaceans, and aquatic plants, in order to enhance production through rearing (Philippine Statistics Authority [PSA], 2022).

Fishing – The term "fishing" describes the act of removing fishery species from their natural habitat or state, either with or without the aid of fishing vessels (*The Philippine Fisheries Code of 1998.*, n.d.).

Marine – Refers to the body of water in the ocean away from the coast, including Manila Bay, the Visayan Sea, etc (Philippine Statistics Authority [PSA], 2022).

Fisheries – All actions associated with fishing, culture, preservation, processing, marketing, development, conservation, and management of aquatic resources and fishery areas, including the right to fish or take aquatic resources thereof (Republic Act No. 8550 otherwise known as "The Philippine Fisheries Code of 1998.") (*The Philippine Fisheries Code of 1998.*, n.d.).

Precision fishing – The practice of maximizing fishing operations and management via the use of cutting-edge instruments and technologies (SafetyNet Technologies, 2021).

Synthetic Image – RGB photos, segmentation maps, depth images, stereo-pairs, LiDAR, and infrared images are all examples of synthetic data used in computer vision. A generative model that resembles the latent space of real-world data is usually used to generate synthetic data for photos and movies (Shah, 2023).

1.5.2 Operational Definition of Terms

Cut, Paste, and Learn – An algorithm that generates synthetic images by pasting foreground images of objects or classes to be detected on top of randomized background images. The algorithm may augment the features of the original image and paste it around the backgrounds randomly (Dwibedi et al., 2017).

Convolutional Neural Networks (CNN) – Also known as ConvNet, a convolutional neural network is an artificial neural network (ANN) that is frequently used to evaluate visual imagery. Since a CNN learns directly from data, laborious feature extraction is not necessary (Yamashita et al., 2018).

Detectron2 - Detectron2 is an open-source software library for object detection, segmentation, and pose estimation written in PyTorch. It was developed by Facebook AI Research (FAIR) and is based on the original Detectron library. It includes implementations of several state-of-the-art object detection algorithms such as Mask R-CNN, RetinaNet, and DensePose. Detectron2 can be used for tasks such as image classification, object detection, semantic segmentation, and instance segmentation. It is designed to be flexible and extensible, allowing users to easily add new models and features (Honda, 2020).

Mask R-CNN – Mask R-CNN (Region-Based Convolutional Neural Networks) is a deep neural network architecture that combines elements of object detection and semantic segmentation from traditional computer vision. It aims to solve instance segmentation problems in computer vision, which are crucial when trying to identify different objects within the same image (He et al., 2020).

Parameter – These are the model's coefficients, and the model itself selected them. This indicates that the algorithm optimizes these coefficients during the learning process (by a predetermined optimization strategy) and outputs an array of parameters that minimize the error. For instance, in a task requiring linear regression, your model might be written as $y=b + ax$, where b and a are the parameters. Initializing those parameters is the only thing you need to do (Yamashita et al., 2018).

Hyperparameter – Hyperparameters are variables that have values predetermined before the model training process begins. The model learning rate, other rules throughout the training process, and final model performance can all be affected by the values supplied for these hyperparameters. Hyperparameters can be categorized as either model hyperparameters, which cannot be inferred while fitting the machine to the training set since they relate to the model selection problem, or algorithm hyperparameters, which in theory have no influence on the performance of the model but affect the speed and quality of the learning process. The topology and size of a neural network are two instances of model hyperparameters (Stanford University, n.d.).

Kernel – The kernel of a convolutional neural network is only a filter that is used to extract features from images. The kernel is an iterative matrix that traverses across the input data, conducts a dot product operation with a subregion of the input data, and outputs the result as a matrix of dot products. The stride value is used by the kernel to move the input data. If the stride value is 2, the kernel will relocate the input matrix's pixels by two columns. In essence, the kernel is utilized to extract from the image high-level features like edges (Stanford University, n.d.).

Weight – The terms parameter and weight are frequently used interchangeably. A neural network's weight parameter alters input data in the network's hidden layers. A network of neurons or nodes makes up a neural network. A set of inputs, a weight, and a bias value are contained within each node. An input is multiplied by a weight value when it enters the node, and the resulting output is either observed or forwarded to the neural network's next layer. The hidden layers of a neural network frequently contain the weights of the network (Stanford University, n.d.).

YOLO – “You Only Look Once” is known by the acronym YOLO. This algorithm identifies and finds different things in an image (in real-time). The class probabilities of the discovered photos are provided by the object identification process which is carried out as a regression problem. Convolutional neural networks (CNN) are used by the YOLO method to recognize items instantly. The approach just needs one forward propagation through a neural network to identify objects, as the name would imply (J. Li et al., 2022).

Python – A high-level, all-purpose programming language. Code readability is prioritized in its design philosophy, which makes heavy use of indentation. Python uses garbage collection and has dynamic typing. It supports a variety of paradigms for programming, including functional, object-oriented, and structured programming (Sonatafy Technology, n.d.).

TensorFlow – A machine learning (ML) and artificial intelligence (AI) software library that is free and open-source. It may be used for many other tasks, although it focuses mostly on deep neural network training and inference (Tensorflow.org, n.d.).

Keras – An open-source software framework that offers a Python interface for artificial neural networks. It also provides the library interface for TensorFlow (Tensorflow.org, n.d.).

Streamlit – A Python-based open-source app framework. It enables us to quickly develop web applications for data science and machine learning. Major Python libraries like sci-kit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, and Matplotlib are all compatible with it. Callbacks are not required with Streamlit since widgets are regarded as variables. Computation pipelines are made easier and faster by data caching. The application is automatically deployed in the shared link when Streamlit detects modifications to the associated Git repository (*Streamlit*, n.d.).

Chapter 2

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter will discuss the studies and literature that are related to the proposed algorithms and web applications that the researchers are developing. Those that were included in this chapter help in familiarizing information that is relevant to the present study and will serve as guidelines to pursue their topic entitled “Aquaria: A Web-Based AI Platform Utilizing Convolutional Neural Networks and YOLOv8 for Precision Fishing, Aquaculture, and Marine Sustainability in the Philippines”.

2.1 CONCEPTUAL FRAMEWORK

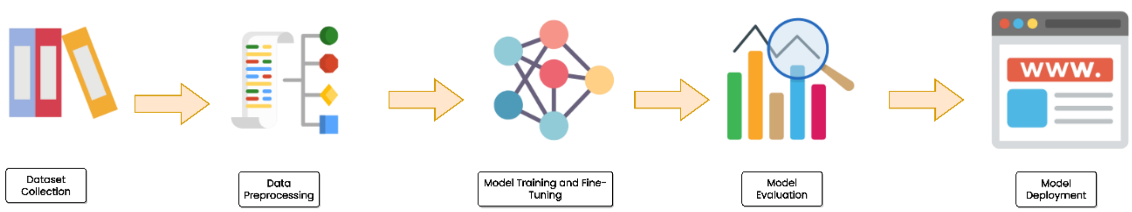


Figure 1. Conceptual Framework

This research unfolded through several interconnected stages, manifesting in the creation of a service that facilitated not just image classification, but also object detection, leveraging the YOLO algorithm. It all began with an extensive data collection phase. A robust dataset was curated from various sources, potentially including public databases or novel data assembled for the study. The quality and diversity of this dataset became pivotal, as it laid the foundation for subsequent research stages.

The raw image data from the collection phase underwent an extensive preprocessing stage next. During this phase, the raw data was transformed into a format suitable for ingestion by the deep learning models. Processes, including but not limited to image resizing, grayscaling, or color normalization, were executed in line with the specific requirements of the model and the attributes of the dataset. Thus, a streamlined, model-ready dataset was born out of raw data, setting the stage for model training.

Model selection and fine-tuning followed data preprocessing. At this juncture, the appropriate deep learning models, one for image classification and the YOLO model for object detection, were chosen and tweaked for the task at hand. Each model's parameters were iteratively adjusted, with the aim to enhance performance and prediction accuracy on the preprocessed data.

Subsequent to the training phase was model evaluation. Here, the performance of the models was assessed against a separate validation dataset. Calculating metrics such as precision, recall, and the F1-score allowed a quantitative understanding of the models' predictive capabilities. This rigorous evaluation was crucial for determining the readiness of the models for deployment and for identifying any areas necessitating further tuning.

The final stage, model deployment, marked the point at which the validated models were put into practical use. This stage considered a variety of factors, including potential users, available computational resources, and the operating environment. At this juncture, the models transformed from theoretical constructs to functional tools, capable of image classification and object detection in the real world.

2.2 RECENT STATE OF FISHERIES IN THE PHILIPPINES

Globally, coastal fisheries are in a condition of constant decline (Pauly et al., 2002). Coastal resource management (CRM) professionals are frequently forced to choose between safeguarding the resources and prioritizing the livelihood of the communities that depend on fishing because of the rising poverty in the fishing villages (Daw et al., 2012). The Philippines is also seeing a deterioration in its fisheries, especially in recent decades due to the rampant exploitation of marine resources. Despite being renowned across the world for its exceptional and productive marine ecosystem, the nation experiences the same problems as other fishing nations (Green, 2003). Below outlines the state of fisheries for the first quarters of the year 2020 – 2022. Among the sector to be considered are the general fisheries state, commercial fisheries, marine municipalities, inland municipalities, and the aquaculture industry.

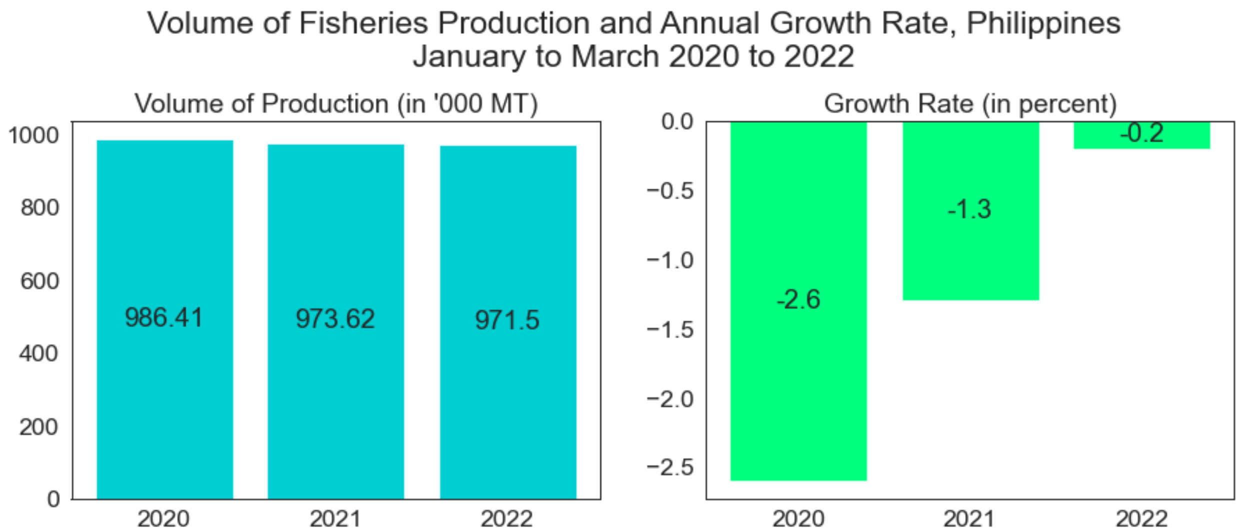


Figure 2. The Volume of Fisheries Production and Annual Growth Rate, Philippines: January to March 2020 to 2022. Source: Philippine Statistics Authority (PSA)

2.2.1 General Fisheries

The total amount of fisheries production was predicted to be 971.50 thousand metric tons in the first quarter of 2022. From the 973.62 thousand metric tons produced in the same quarter the year before, it fell by -0.2%. Commercial and municipal marine fisheries both saw declines in productivity (Philippine Statistics Authority [PSA], 2022). Figure 2 illustrates the volume of Fisheries Production and Annual Growth Rate in the Philippines: January to March 2020 to 2022

2.2.2 Commercial Fisheries

In the first three months of 2022, commercial fisheries produced 177.17 thousand metric tons, which was -8.0 percent less than the 192.67 thousand metric tons produced in the corresponding period of the prior year. The output of the subsector accounted for 18.2% of the total fisheries production (Philippine Statistics Authority [PSA], 2022).

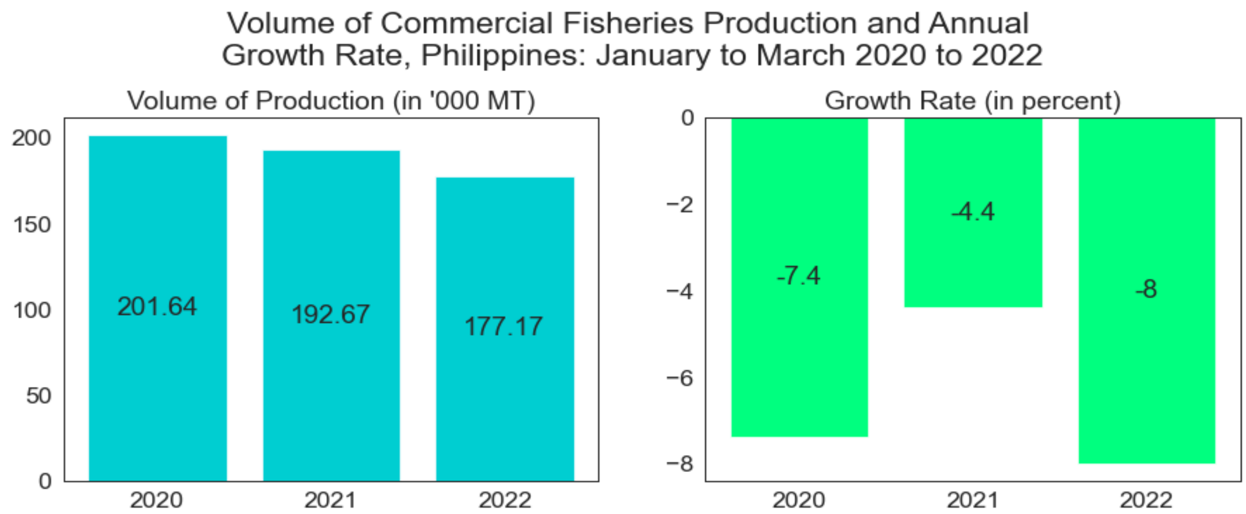


Figure 3. The Volume of Commercial Fisheries Production and Annual Growth Rate, Philippines: January to March 2020 to 2022. Source: Philippine Statistics Authority (PSA)

2.2.3 Marine Municipal

During the quarter, the maritime municipal fisheries subsector reported discharging 218.73 thousand metric tons in total. The volume was 0.9% below the 220.68 thousand metric ton level from the first quarter of 2021. The subsector contributed 22.5 percent of the total fisheries production (Philippine Statistics Authority [PSA], 2022).

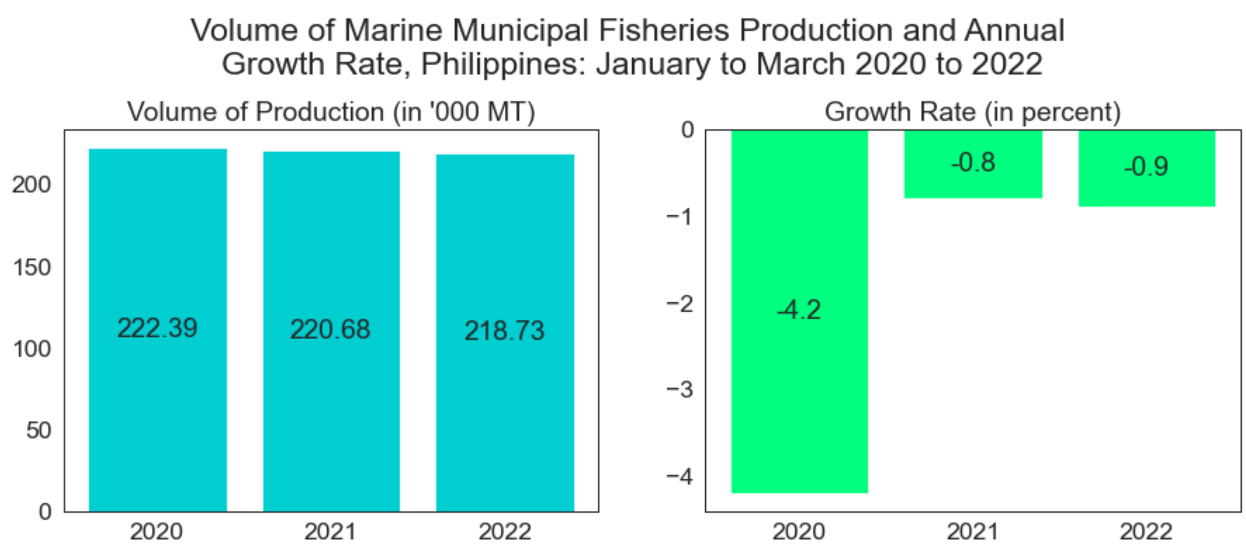


Figure 4. The Volume of Marine Municipal Fisheries Production and Annual Growth Rate, Philippines: January to March 2020 to 2022. Source: Philippine Statistics Authority (PSA)

2.2.4 Inland Municipal

Production of municipal inland fisheries increased during the quarter by 4.7%. The output amount was estimated at 37.31 thousand metric tons the year before but was recorded to be 39.05 thousand metric tons. Municipal inland fisheries generated 4% of the overall fisheries production (Philippine Statistics Authority [PSA], 2022).

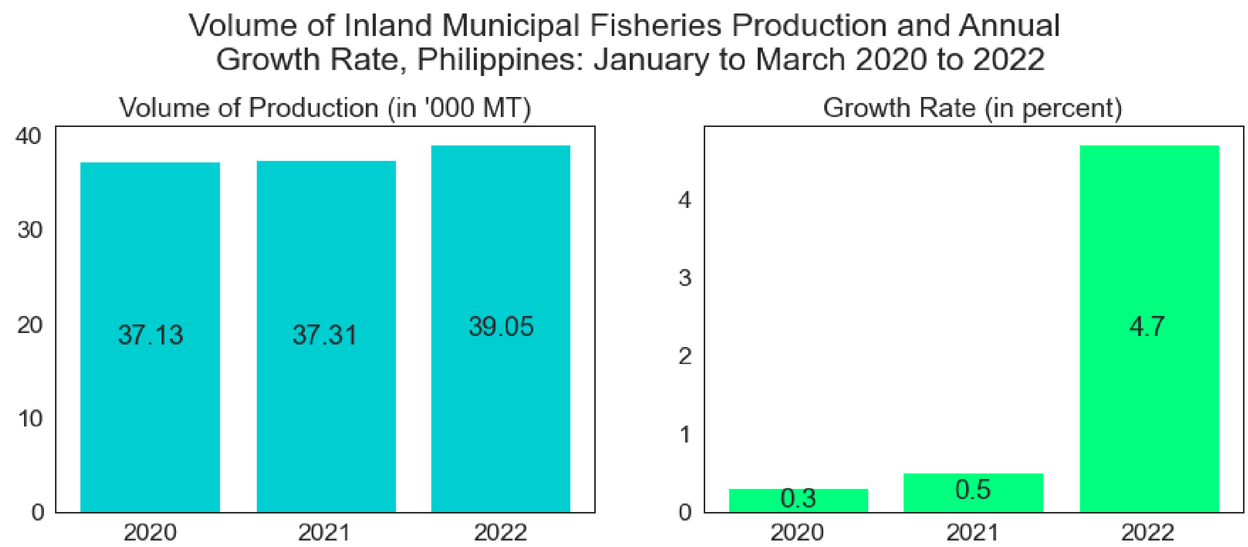


Figure 5. The Volume of Inland Municipal Fisheries Production and Annual Growth Rate, Philippines: January to March 2020 to 2022. Source: Philippine Statistics Authority (PSA)

2.2.5 Aquaculture

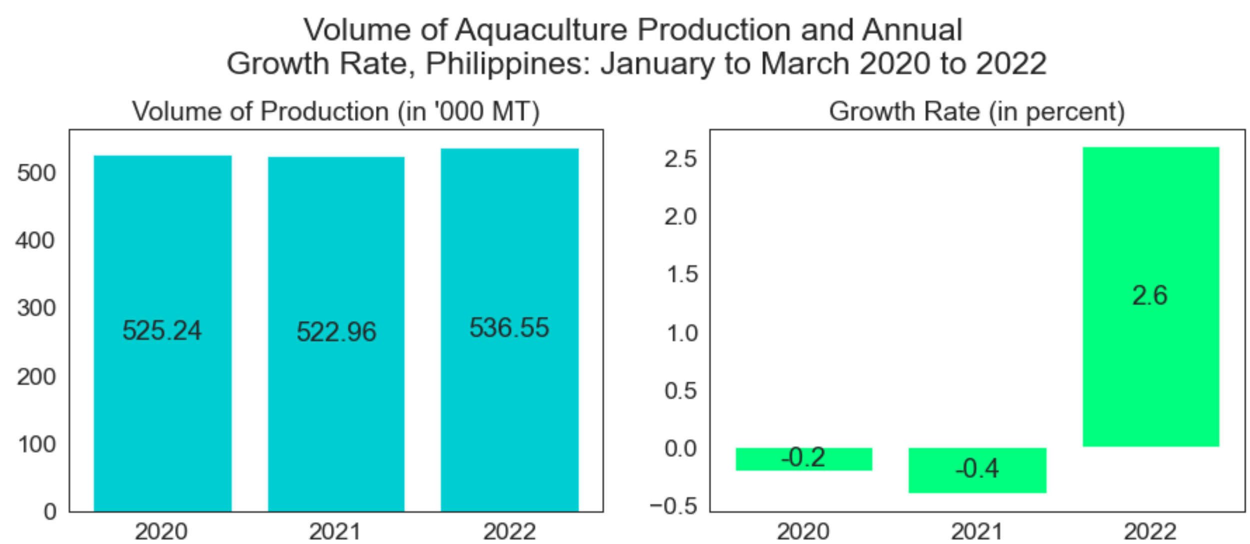


Figure 6. The Volume of Aquaculture Production and Annual Growth Rate, Philippines: January to March 2020 to 2022. Source: Philippine Statistics Authority (PSA)

Total aquaculture farm harvests climbed by 2.6 percent to 536.55 thousand metric tons in the first quarter of 2022 from 522.96 thousand metric tons in the same quarter of the previous year. Of all fisheries production, aquaculture accounted for 55.2%. (Philippine Statistics Authority [PSA], 2022).

2.3 FISHING METHODS AND TECHNIQUES IN THE PHILIPPINES

Around the world, many fishermen continue to capture fish using conventional, somewhat low-tech methods because fishing is mostly done for food and subsistence (Alimen et al., n.d.). In the Philippines, local fishermen also have their ways of fishing which they use to sustain their everyday livelihood. In an interview conducted by (Alimen et al., n.d.) with the Fisher Folks of San Jose Antique, they were able to gather data about the methods many of the local fishermen use to catch fish. They choose to employ hooks (taga), nets, and spears (pamana) to catch fish within the coastal areas of their vicinity. Additionally, fishermen have employed commercial fishing vessels outfitted with cutting-edge equipment to aid their livelihood. The next subsections will further discuss the varieties of traditional fishing methods used by Filipino fishermen as well as the techniques used to lure fish and maximize their catch.

2.3.1 Fishing Methods

One of the innovative livelihoods practiced by Filipinos is fishing. Different sorts of fishing techniques and equipment were created and used by fishermen in coastal areas. A total of 31 fishing methods were recorded (Joaquin, 2021) and were divided into passive, active, pot, and hand fishing. In passive fishing, fishermen set up their equipment on the sea, river, or estuary and rely on the flow of the water to draw fish to them. In contrast, fishermen that are actively fishing use gear to pursue and catch their intended species (Joaquin, 2021). In pot fishing, the fishermen employ a variety of netting-covered cages, including turtle-shaped, box-shaped, and frustum-shaped ones. A pot may contain one or many entrances to let specific organisms in, depending on the design. The entry can be a straightforward slit, a funnel-shaped opening, or a non-return valve (Bucol, 2016; Joaquin, 2021; Kawamura & Bagarinao, 1980). Lastly, the most rudimentary form of fishing, known as hand fishing, involves groping or catching fish just with one's bare hands. The corresponding local methods per fishing category are outlined in Table 1.

Figure 7. Fishing Methods in the Philippines

(A) Spear Fishing (B) Rod Fishing (C) Net Fishing (D) Pot Fishing

2.3.2 Fishing Techniques

The effectiveness of a fisherman's skills depends on his or her understanding of fish behavior and habitat. As a result, understanding the fishing location and how the fisherman sets the traps is regarded as a crucial and sophisticated talent. The *kurantog* is one method used by fishermen to draw in fish. Using a pulse stick or wooden paddle, fishermen make noise to entice fish or get them to move (Joaquin, 2021). The technical modification of fishing gear, notably for the design of trap entrances, also reflects the complexity of the fishing techniques. The design is adjusted to the local environmental conditions, taking into account the local fish behavior and natural factors such as tidal movements (Bucol, 2016; Joaquin, 2021; Kawamura & Bagarinao, 1980). The approach for positioning the gear must consider the typical flow of the stream because the quantity of fish caught depends on it. As a result, adjustments are made. The season also affects how these stationary gears are built and destroyed.

Table 1. *Traditional Fishing Methods in the Philippines. Source: (Joaquin, 2021)*

Passive Fishing	Active Fishing	Pot Fishing	Hand Fishing
Arong	Hudhud	Bobo	Pagpangaret
Bira-bira	Pagpangasag	Timing	Pamatad
Padugmon	Pagataw	Panggal	Panihi
Pagpamangrus	Paiwag		Pamuho
Taba	Pagsibot		Pangsisi/Panisi
Tangab	Palumoy		Panikop
Taon	Pamana		Panulo
Pamintol	Pamanti		
Pamunit	Pukot		
Batak-batak	Pamusit		
Saluran			

2.4 IMPACTS OF MARINE POLLUTION ON THE PHILIPPINE FISHERIES

According to some estimates, one of the world's major sources of marine plastic pollution is the Philippines (Jambeck et al., 2015; Lebreton et al., 2017). Several rivers in the Manila metropolitan region are thought to be the main channels for land-based plastic garbage to enter the ocean (van Emmerik et al., 2020). The persistence of plastics in the ocean and their negative effects on marine life and possibly human health have made them a serious environmental issue (Lebreton et al., 2017). Numerous species have reportedly perished as a result of the negative consequences of these marine contaminants (Abreo, n.d.; Bergmann, 2015). In the fisheries sector, the nutrition and quality of the water can be harmed by pollution, which in turn, can limit fish development. In addition to causing different fish diseases, an increase in harmful material contained in the water, such as nitrogen and phosphorus, will also enhance the likelihood that fish will become infected. Hence, fish breeding and reproduction will be severely limited (Yuan et al., 2021).

In the study conducted by (Waldichuk, 1974), there are seven ways marine life may be damaged by pollution: (1) habitat loss; (2) acute poisoning by toxic wastes; (3) detrimental water quality alteration; (4) sub-lethal effects of pollutants impairing nutrition, development, migration, resistance to disease and parasites, and interference with reproduction; (5) bacterial and viral contamination; (6) bioaccumulation of toxic metals and organic chemicals; and (7) tainting and/or staining of the meat by organic and/or metallic substances. Through ecological and commercial dynamics, accumulated pollution and economic expansion have an impact on the profitability of the fisheries industry (Bergland et al., 2019). Many of the productive coastal regions would be lost to fisheries despite the most active conservation programs, supported by strict national legislation. Fisheries demands will either be outweighed by national priorities that are unrelated to marine conservation, or expenses will be too high to support comprehensive restoration (Bergland et al., 2019; Waldichuk, 1974).

Many of pollution's consequences are undetectable and covert. The more catastrophic long-term deficits in fish stocks are brought on by indirect effects like predation, disease, and poor reproduction, as opposed to direct population eradication, such as fish poisoning. Fish that experience unnatural stress at any point in their lives may lose the ability to carry out the tasks required to complete their life cycle. A fish may not breed and leave no natural way for the species to survive if pollution makes it difficult for it to locate its natal stream (Waldichuk, 1974). To take decisive action to meet the challenge, it is important to fully comprehend the potential harm caused by pollution. In general, (Yuan et al., 2021) argue there are two approaches to dealing with environmental pollution: "adaptation" and "mitigation." First, in terms of "adaptation," there needs to be more attention on research, development, and the promotion of pertinent technologies (such as the development of purification equipment, novel varieties, and environmental pollution warning mechanisms). Second, in terms of "mitigation," innovation should be actively encouraged and new methods for dealing with urban wastewater should be investigated to address the environmental damage caused by pollutants; on the other hand, improvements are required in the chemical products for daily use, such as liquid detergent and toilet cleaners, etc., and more pollution-free products should be developed.

2.5 PRECISION FISHING AND AQUACULTURE

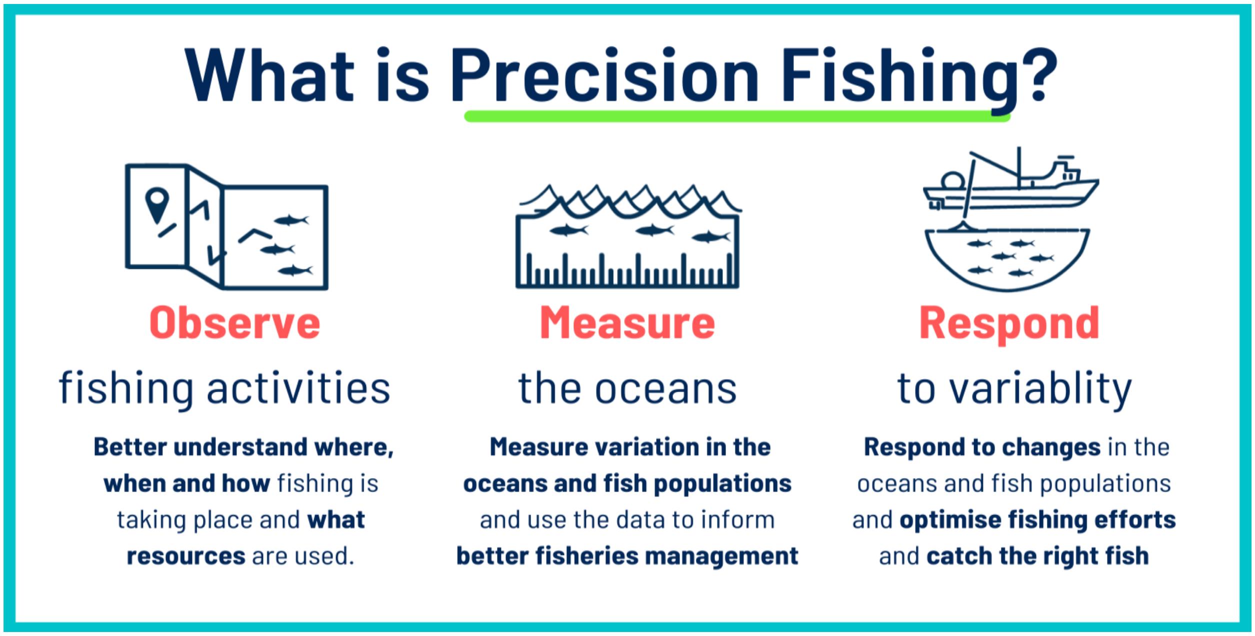


Figure 8. Precision Fishing and its processes. Source: (SafetyNet Technologies, 2021)

The employment of cutting-edge equipment and technology to control and optimize fishing operations is known as precision fishing (Costello et al., 2016). The term "precision" refers to the ability of these instruments to monitor fish stocks in almost real-time, catch the appropriate fish, and carry out the proper interventions to safeguard fish stocks and the marine ecosystem. This can be done by utilizing technology to monitor, assess, and react to variations in commercial fish populations (SafetyNet Technologies, 2021). To maximize your fishing efforts, you must first examine when, where, and how fishing is occurring, as well as what is being caught. Reliable and consistent statistics on fishing effort, bycatch, and regulatory compliance are necessary for effective fisheries management. This information has typically been gathered by paper logbooks, impartial observers, and questionnaires. The main source of this information has been independent observers on fishing boats. However, because these observers are pricey, many fisheries do not have the observer coverage required to make sensible management choices. Additionally, these data gaps in fishing must be closed for precision fishing to function (SafetyNet Technologies, 2021).

Rapid and significant changes have occurred in how recreational anglers engage with fisheries resources because of technology created for or adopted by the recreational fisheries sector (such as fishermen and the recreational fishing industry). Technology is fundamentally altering every aspect of recreational fishing, from advances in fish detection and capture to simulating their natural prey and reaching previously inaccessible waterways to fishermen sharing their experiences with others (Cooke et al., 2021). For millions of individuals, recreational fishing is a significant leisure activity. Through direct harvesting and the unexpected mortality of released fish, recreational fishing has the potential to affect fish abundance as well (Arlinghaus & Cooke, 2005; Cooke et al., 2021). Aside from the fisheries sector, precision aquaculture has also its fair share of development. Aquaculture may now integrate the Internet of Things (IoT) due to recent technological advancements. Modern farms are made up of hundreds of interconnected sensors that communicate with one another, store and serve data, and connect to a fog and cloud network (O’Donncha & Grant, 2019). Aquaculture's future development will further focus on the following areas: ecological, facility, industrial, and intelligent (O’Donncha & Grant, 2019; Wang et al., 2021).

2.6 APPLICATIONS OF COMPUTER VISION IN FISHERIES

Computer Vision has been widely integrated within the fisheries industry to improve the existing mechanisms and tools used to maximize efficient fish collection. Automated systems for classifying, sorting, and processing fish and fish products use computer vision algorithms to aid in the processes used to lessen human manual labor (Mathiassen et al., 2012). Additionally, fisheries, fish farming, and fish processing are understood and improved with the help of computer vision. It can be used in fish farming for a variety of purposes, including fish target identification, fish measuring, and behavioral monitoring (Liu et al., 2014; Yang et al., 2021). However, despite the intersection between the field of deep learning and fisheries showing promising potential for innovative precision fishing, there are still many considerations that need to be addressed. Although fish farming applications for computer vision are expanding and demonstrating good development dynamics, they are still relatively specialized. Therefore, strategies for modifying current computer vision solutions to the circumstances of fish farms in various places will need to be developed in the future (Petrov & Popov, 2020).

2.6.1 Computer Vision for Fish Feeding

Future aquaculture breeding will rely heavily on intelligent feeding and establishing this system will become a top priority. Simultaneously, computer vision is gradually becoming a viable intelligent feeding automation algorithm for precision fish farming and turning everyday routine tasks into a sequence of data-reliant conditions (Liu et al., 2014). Aquaculture will grow as a result of the two systems working together, and their further development will be based on new ideas and the application of existing techniques (An et al., 2021). Moreover, developing a robust fish-feeding system requires underwater image preprocessing. It aids in the identification and behavioral observation of fish, enabling more precise calculation of the quantity and timing of feeding. Preprocessing underwater fish images is a crucial component of many fish target selection or development techniques (An et al., 2021; C. Zhou et al., 2018). Computer vision, a key component of artificial intelligence, is now included in every element of smart feeding (An et al., 2021). In addition to increasing productivity and product quality, computer vision technology can significantly minimize the need for labor.

2.6.2 Computer Vision for Measuring Fish Parameters

When managing aquariums and fish farms or evaluating fishways, computer vision algorithms can take the place of more conventional methods like direct observation, which are either impracticable or have an impact on fish behavior (Rodriguez et al., 2015). A more effective option may be computer vision, which is a quick, affordable, consistent, objective, and non-destructive inspection technique (Cappo, Harvey, Malcolm, et al., n.d.). Computer vision algorithms have also been successfully used in fish studies and underwater environments. The employment of acoustic transmitters and a video camera to observe the behavior of diverse species are early examples of these applications (Cappo, Harvey, & Shortis, n.d.). Computer vision algorithms have also been successfully used in fish studies and underwater environments. The employment of acoustic transmitters and a video camera to observe the behavior of diverse species are early examples of these applications (Cappo, Harvey, & Shortis, n.d.). More recently, several computer vision algorithms have been employed to analyze the swimming patterns of fish, classify the species of fish, measure fish size, and observe fish behavior (Petrell et al., 1997; Rodríguez et al., 2015; Tonachella et al., n.d.; Zion et al., 1999).

2.6.3 Computer Vision for Monitoring Fish Behavior

Examining a fish's appearance or behavior can reveal a wealth of information about the species, including its health and development as well as its relationships with the ecosystem. It can be used to forecast the best developmental stage for potential commercial exploitation, estimate growth rates, and provide early warning signs of health issues (Rodriguez et al., 2015). Fish farming is a labor-intensive process that takes a lot of time. Lack of monitoring results in fish loss, hence automatically monitoring fish farms would lessen the risk of predicament and ensure a healthy development cycle within the environment. In recent times, Fish Farm makes an automatic effort to spot any anomalies in fish ponds, like fish with atypical behavior, which lowers the risk of fish mortality and boosts fish productivity (Keerthi & Subhashini, 2022). This is one of the innovative approaches developed using the capabilities of Computer Vision, especially within the intersections of fisheries and aquaculture. Furthermore, the ability to better observe fish behavior has been made possible by recent advancements in computer vision technology. With the aid of such technology, inspection instruments can be non-destructive, quick, affordable, reliable, and objective and an evaluation approach based on image analysis and processing can be used in a wide range of applications (Niu et al., 2018).

2.7 SYNTHETIC IMAGE DATA GENERATION

Lack of labeled data is a typical issue when training Convolutional Neural Networks (CNNs). Situations, where a dataset of tagged photos has never been developed, are where this issue is most prevalent (Arcidiacono, n.d.). Gathering and annotating photos from the actual world requires significant time and financial inputs, and is typically too passive to create datasets with particular qualities, including small object areas and high occlusion levels (Arcidiacono, n.d.; Tian et al., 2017). Synthetic image production appears to be the only option in this situation. In industrial settings, the objects belonging to a class that needs to be recognized are frequently indistinguishable. Because the problem of intra-class variability will be less severe in this use case than in other use cases, synthetic data generation approaches may produce better outcomes. The generation of synthetic image data for Computer Vision tasks has been widely used in many different studies across multiple domains (Chen et al., 2021; Frolov et al., 2022; Manettas et al., 2021). Several methods have been developed to automatically create synthetic images. These techniques can be divided into two broad categories: those that generate images from CAD models using an approach similar to that given by (Peng et al., 2015) and those that generate images by cutting an object out of one image and pasting it onto another using an approach similar to that presented in (Dwibedi et al., 2017).

2.7.1 Computer-Aided Design (CAD) Models

Many different sectors employ three-dimensional (3D) computer-aided design (CAD) model reconstruction techniques, including free-viewpoint video reconstruction, robotic mapping, tomographic reconstruction, 3D object identification, and reverse engineering. Researchers are looking into the reconstruction of 3D CAD models using learning-based techniques as deep learning techniques continue to advance in many different domains of sturdy (Lee et al., 2021). Promising results have been obtained when using synthetically generated data from CAD models to enable object detection in the real environment (Kohtala & Steinert, 2021). Numerous methodologies and approaches were used to leverage the capabilities of the synthetic data within different use cases and research interests. The fundamental problem with using openly accessible CAD models for training is that they usually lack other low-level cues like object texture, backdrop, realistic stance, lighting, etc. while accurately capturing the 3D geometry of the object (Tian et al., 2017). It is necessary to have CAD models and encoding of the physical characteristics of an object, such as its color and reflectance, to use the technique for creating CAD models (texture). It's not always simple to meet this condition (Arcidiacono, n.d.). More recently, (Liebelt & Schmid, 2010; Stark et al., 2010; Sun et al., n.d.) exclusively employed 3D CAD models as their source of labeled data and restricted their research to a few areas like cars and motorbikes. (Pinto et al., 2011) used simulated data to investigate feature invariances for SIFT, SLF, and other features.

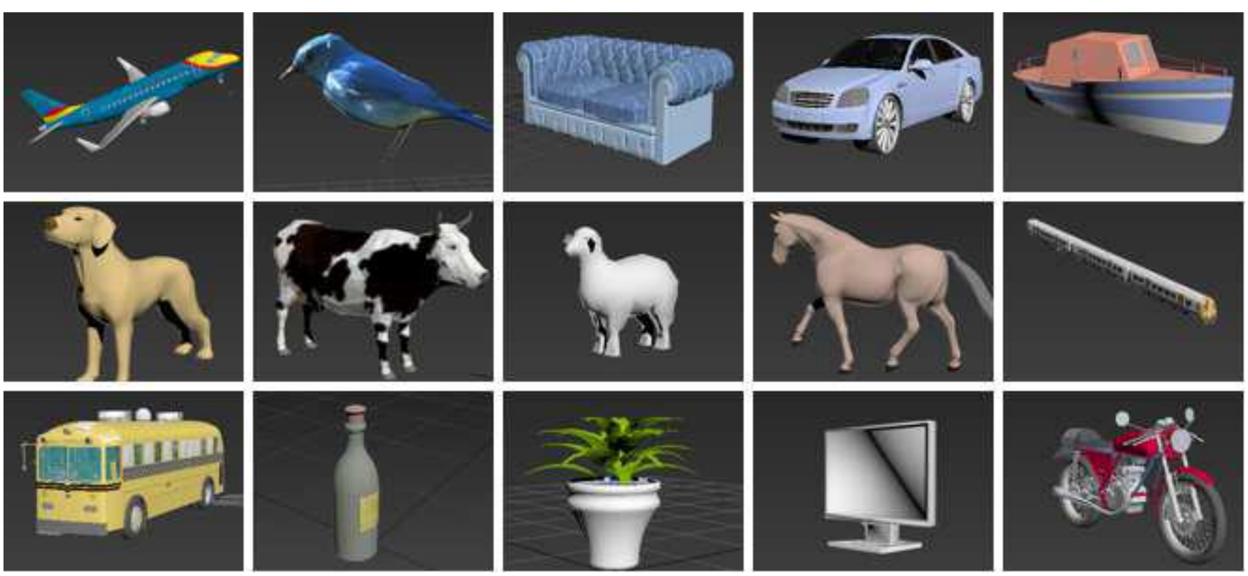


Figure 9. CAD Synthetic Images used in the study conducted by (Peng et al., 2015)

2.7.2 Cut Paste Data Generation

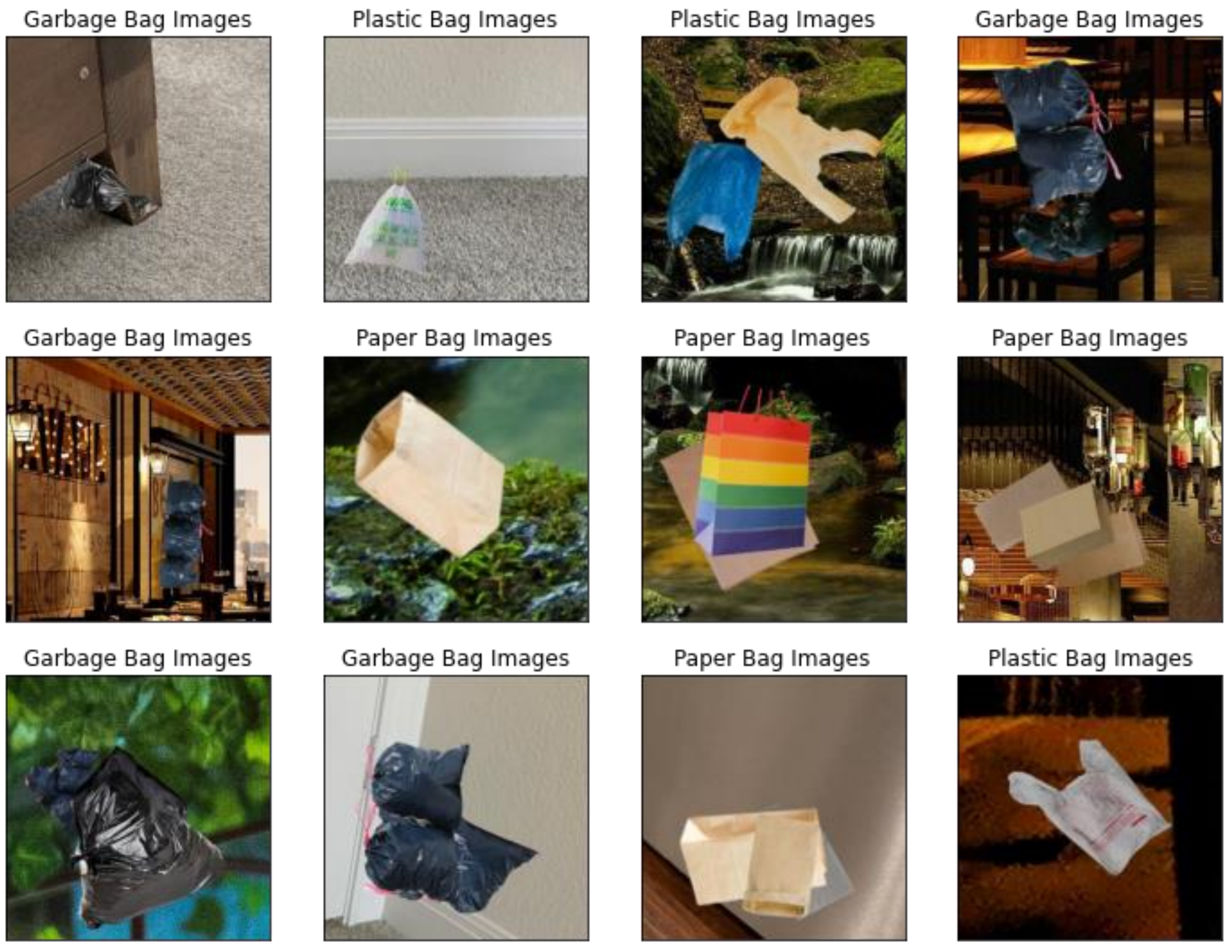


Figure 10. Synthetic Disposable Bag Images generated using Cut-Paste Algorithm

Due to the numerous parameters that must be learned, CNN-based object detectors need a lot of labeled data to train on. The training data for object instance detection should also include information on how the object is seen from different angles and other bothersome factors like occlusion, clutter, and poor lighting. It takes a lot of time and money to manually gather and annotate scenes with the characteristics. The frequently limited ability of trained models to generalize across many environments and backgrounds is another consideration in annotating (Georgakis et al., 2017). The concept of Cut-Paste and Learn image synthesis was thoroughly discussed in the study conducted by (Dwibedi et al., 2017). They presented a simple technique to synthesize annotated training images for instance detection. The great thing about the algorithm is it automatically generates annotations based on the positioning and metadata of the pasted foregrounds within different backgrounds. Therefore, the researchers won’t have to worry about the data gathering itself but instead focus on adjusting the parameters to invoke realism within the dataset. In the study conducted by (Georgakis et al., 2017) they focused on superimposing item instances at various dimensions and positions into real scenes while minimizing the disparity in brightness and taking advantage of the right setting. This approach could also be merged with a real dataset as an upscaling component to further improve the performance of the existing Computer Vision modes.

2.8 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are a subclass of Artificial neural networks (ANNs) that have gained prominence in several computer vision applications (Yamashita et al., 2018). They are gaining interest in different fields including fisheries (Allken et al., 2019; French et al., 2020; Rekha et al., 2020). Using a variety of building pieces, including convolution layers, pooling layers, and fully connected layers, CNN is intended to automatically and adaptively learn spatial hierarchies of features through backpropagation (Stanford University, n.d.). Like regular neural networks, convolutional neural networks are composed of neurons with trainable weights and biases. Each neuron processes several inputs conducts a dot product and may optionally do a non-linearity as a follow-up. From the unprocessed image pixels at one end to the class scores at the other, the entire network continues to represent a single differentiable score function. As for the final (fully connected) layer, they still have a loss function (such as SVM/Softmax), and all the learning strategies for conventional neural networks still hold (Stanford University, n.d.). A simple ConvNet consists of a series of layers, each of which uses a variational function to morph one volume of activations into another.

2.8.1 Layers

Convolutional Layer, Pooling Layer, and Fully Connected Layer are the three primary types of layers that we employ while creating ConvNet systems (exactly as seen in regular Neural Networks). These layers will be stacked to create a complete ConvNet architecture (Stanford University, n.d.). From the initial pixel values to the final class scores, ConvNets alter the original image layer by layer. It should be considered that certain layers have parameters and others do not. More specifically, the CONV/FC layers carry out transformations that depend on the parameters as well as the activations in the input volume (the weights and biases of the neurons). On the other hand, the RELU/POOL layers will carry out a fixed function. To ensure that the class scores that the ConvNet computes are consistent with the labels in the training set for each image, the parameters in the CONV/FC layers will undergo gradient descent training (Stanford University, n.d.).

2.8.2 Non-linear Activation Function

Every kind of activation function in every kind of neural network is to connect the input to the output. By calculating the weighted summation of the neuron input and its bias, the input value is derived (if present). This means that by producing the corresponding output, the activation function decides whether or not to fire a neuron in response to a specific input (Alzubaidi et al., 2021). The most popular nonlinear activation function employed today is the rectified linear unit (ReLU), which computes the function: $f(x) = max(0, x)$. Smooth nonlinear functions, such as the sigmoid or hyperbolic tangent (tanh) function, were once widely used since they are mathematical representations of a biological neuron action (Yamashita et al., 2018).

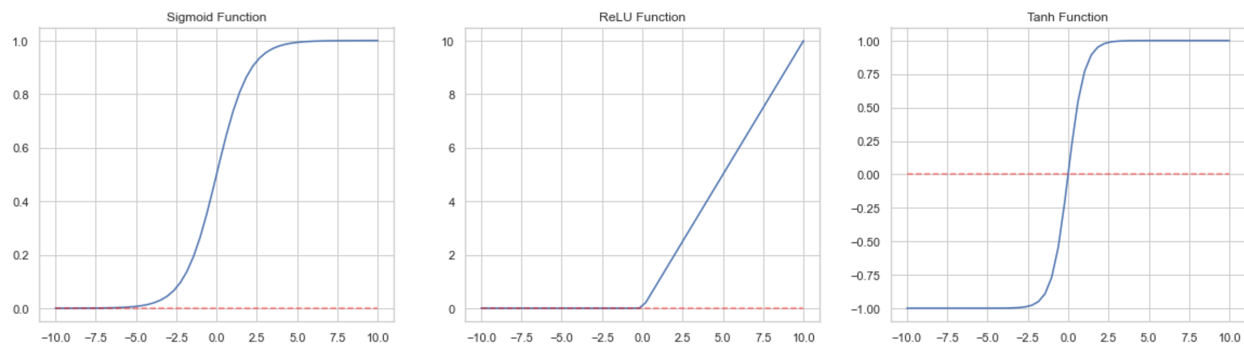


Figure 11. Graphs of Common Activation Functions used in Deep Learning

2.8.3 Gradient Descent

The direction with the biggest changing value at one point is referred to as the gradient, which is a vector in mathematics. Gradient Descent is a machine learning technique that defines the difference between two variables in training neural networks (X. Zhou, 2018). We can determine the direction of the weight vector shift that will result in the best results and is mathematically guaranteed to be the direction of the sharpest decline (at least in the limit as the step size goes towards zero). The gradient of the loss function will have an impact on this direction (Stanford University, n.d.). The slope of a one-dimensional function is the function's instantaneous rate of change at any potential interest point. For functions that accept a vector of numbers rather than a single number, the gradient is a generalization of the slope. The gradient is also essentially a vector of slopes for each dimension in the input space, often known as derivatives (Stanford University, n.d.). The mathematical formula for a 1-D function's derivative regarding its input is:

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h} \quad (1)$$

We refer to the derivatives as partial derivatives when the functions of interest accept a vector of numbers rather than a single number, and the gradient is essentially the vector of partial derivatives for each dimension. Gradient Descent is the process of repeatedly evaluating the gradient and then updating the parameters after computing the gradient of the loss function (Stanford University, n.d.). Its default appearance is as follows:

```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

All libraries for neural networks are built around this straightforward loop. Gradient Descent is currently by far the most popular and well-established method of optimizing Neural Network loss functions, but there are other techniques to carry out the optimization (such as LBFGS) (Stanford University, n.d.).

2.8.4 Backpropagation

The method of progressive computing used by nonlinear multilayer neural networks is referred to as backpropagation (Nguyen et al., 2020). Backpropagation is among the supervised learning and multi-layered training programs that make use of errors when modifying the weight value both in the forward and backward propagation processes (Izhari et al., 2020). During the feed-forward computation of neural networks, the result output value is not near the target or teacher output value. There is a difference between target, and actual feed forward values lead error value. The neural network model tries to give the best prediction output will good tolerance for that we have to minimize the error rate. This can be done with Backpropagation (Sekhar & Meghana, 2020).

To determine the required weight and bias of the neural network, such a method uses a collection of input and output values. Traditional BP networks, however, have numerous drawbacks, including slow convergence and a simple fall to the local minimum (H. Wu et al., 2016). Therefore, several generalization techniques like Bayesian regularization (BR) and Levenberg-Marquardt (LM), which have the advantage of having a smaller mean squared error, are used to reduce the error associated with the backpropagation procedure (Burden & Winkler, 2008; Saini, n.d.). For instance, the research conducted by Kayri, 2016 revealed that BR outperformed LM (Kayri, 2016; Nguyen et al., 2020). Although gradient descent is a backpropagation training technique, its convergence rate is slow. Therefore, the backpropagation training network, by the Bayesian regularization method, is one of the algorithms that enhances the convergence or learning rate of the neural network (Pan, 2014; R. Zhou et al., 2018).

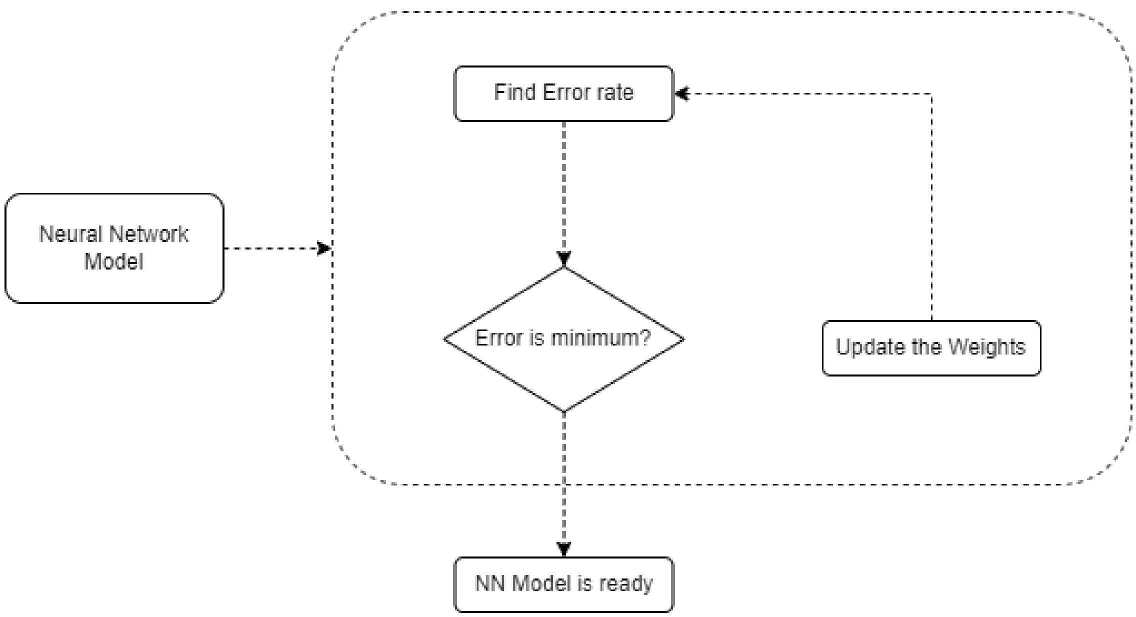


Figure 12. Flow diagram of the working mechanism of backpropagation.

Source: (Sekhar & Meghana, 2020)

2.8.5 Transfer Learning and Fine Tuning

It is widely known that deep learning systems need a lot of data to be trained (Brigato & Iocchi, 2020; Romero et al., n.d.). Data collecting, however, is extremely expensive, if not completely impractical. It is quite uncommon to have a dataset large enough to train a whole convolutional network from the beginning (with random initialization), hence this is rarely done. It is more typical to train a ConvNet on a very big dataset (like as ImageNet, which has 1.2

million images with 1000 categories) and then utilize the ConvNet as an initialization or a fixed feature extractor for the relevant job (Stanford University, n.d.). One can get around the limited data issue by applying models that have been trained on a large dataset to the target domain with a small dataset (Ramdan et al., 2020). Through the reuse of previously learned features when datasets are insufficient, transfer learning plays a significant role in addition to parameter initialization(Romero et al., n.d.). Transfer learning, which aims to transfer knowledge from source tasks to a target task, is an effective solution. Fine-tuning is a popular transfer learning technique for deep neural networks where a few rounds of training are applied to the parameters of a pre-trained model to adapt them to a new task (Guo et al., 2020).

2.9 IMAGE CLASSIFICATION

The most crucial and challenging task in the realm of computer vision is classification. The description, texture, or likeness of objects or entities serve as the basis for classification. The categorizing of photographs into one of several specified categories is known as image classification. An image is represented by units called pixels. Pixels in an image are categorized into various classes. Image acquisition, picture pre-processing, and image segmentation are all included in image categorization (Ponnusamy et al., 2017). To categorize an object into the right category, the Image Classification system compares predetermined patterns in a database with the object. In many different application fields, including remote sensing, vehicle navigation, biomedical imaging, video surveillance, biometry, industrial visual inspection, robot navigation, and vehicle navigation, image classification is a critical and difficult issue (Ponnusamy et al., 2017).

Many studies have also found image classification within fisheries (Dhar & Guha, 2021). In the study conducted by (Qiu et al., 2018) they demonstrated that b low-quality and small-scale fine-grained fish image data can be handled by improving transfer learning and squeezing and excitation networks Using five datasets, the enhanced transfer learning technique outperforms the present CNN models in classifying fish images. On the other hand, (Pandiyan et al., 2020) used the SVM algorithm, which has four kernel functions: linear, polynomial, sigmoid, and radial basis functions. The foundation of numerous fish image applications, including illness early warning and diagnostics, animal behavior, aquatic product processing, etc., is the extraction of fish contours from photographs. Yao et al., 2013 proposed a new fish image segmentation approach that combines the K-means clustering segmentation algorithm and mathematical morphology to increase the precision and stability of fish image segmentation. Many other studies have been conducted utilizing image classification techniques in the fisheries sector. For more reference about this, consider reading section 2.6, *applications of computer vision in fisheries.*

2.10 INSTANCE SEGMENTATION AND OBJECT DETECTION

Instance segmentation has emerged as a crucial field of study in computer vision in recent years. This innovation has been used in a variety of contexts, including robotics, healthcare, and autonomous driving. Technology for instance segmentation not only recognizes the item's location but also identifies its edges for each instance, which can address both object detection and semantic segmentation happening at once (Sharma et al., 2022). On the other hand, a bounding box is generated around an identifiable object after it has been precisely localized and recognized. Object detection classifies one or more objects in a picture by combining localization and identification (Kumar, 2021).

2.10.1 Mask R-CNN

The process, known as Mask R-CNN, expands Faster R-CNN by including a branch for object mask prediction in addition to the existing branch for bounding box identification. Mask R-CNN runs at 5 frames per second, adds only a little overhead to Faster R-CNN, and is easily trainable. Mask R-CNN is also simple to generalize to other problems, enabling us, for example, to estimate human poses within the same framework (He et al., 2020). Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI (He et al., 2020). Some studies cover the applications of Mask R-CNN within the fisheries sector. In the study conducted by (Siddagangu et al., 2021), they proposed a system for classifying different species of fish. The algorithm suggests detecting and classifying the fish species Albacore, BigEye, YellowFin, Moon, Dolphin, and Shark. The system offers 93% accuracy. On the other hand, the study conducted by (Chang et al., 2021) utilizes the contrast between fish and the background in sonar photos taken from various shallow seas can vary greatly. Furthermore, to offer "standardized" feature maps for Mask R-CNN and to make it easier to transfer Mask R-CNN trained for one fish farm to the others, a preprocessing convolutional neural network (CNN) was proposed. According to experimental findings, Mask R-CNN on PreCNN output is more accurate than Mask R-CNN applied directly to sonar images. It is also more efficient to apply Mask R-CNN plus PreCNN trained for one fish farm to new fish farms (Chang et al., 2021).

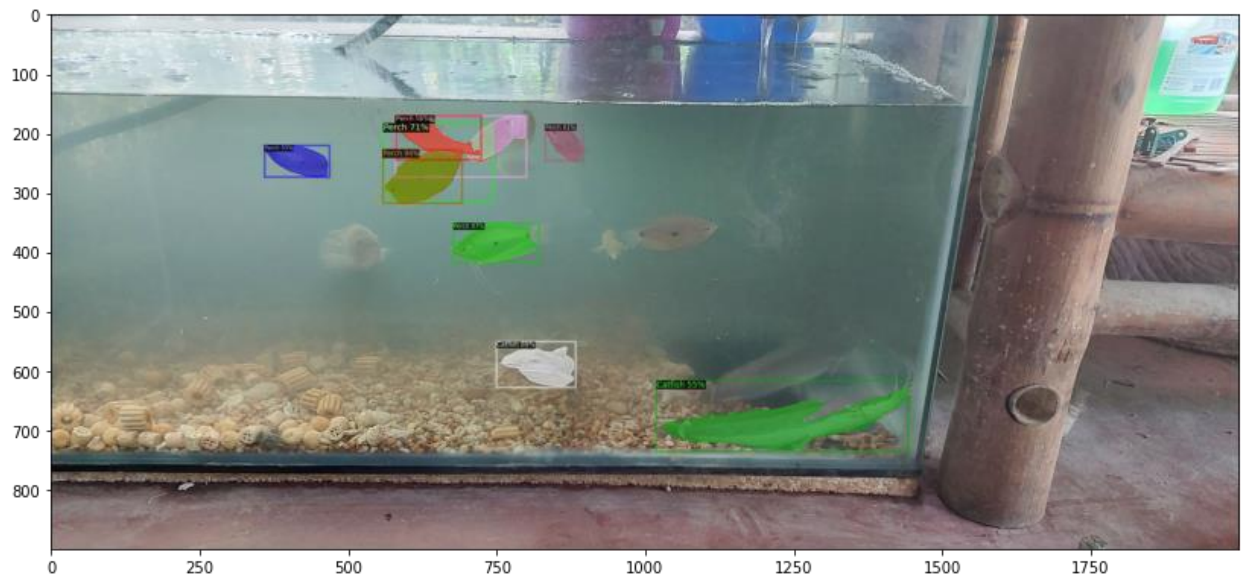


Figure 13. Fish Species Instance Segmentation using Mask R-CNN

2.10.2 Detectron 2

A complete overhaul of Detectron that began with mask can-benchmark is called Detectron2. Detectron2 now has a new, more modular design that makes it adaptable, expandable, and capable of offering quick training on one or more Graphics Processing Unit (GPU) servers. High-quality implementations of cutting-edge object identification methods are included in Detectron2, including DensePose, panoptic feature pyramid networks, and several iterations of the ground-breaking Mask R-CNN model family, which was also created by Facebook AI Research (FAIR). Modern research projects can easily be implemented using it without having to fork the entire software because of its expandable nature (Y. Wu et al., 2019). Detection 2 had been used in many different studies across different domains. In the study conducted by Pham et al., 2020, they utilized the algorithm for segmenting and detecting road damage. The Global Road Damage Detection Challenge 2020, A Track in the IEEE Big Data 2020 Big Data Cup Challenge dataset was used by authors as a testing ground for these methodologies. The findings demonstrate that the X101-FPN basic model for Faster R-CNN using Detectron2's default parameters is effective and sufficiently universal to be applied to many nations in this challenge. With this strategy, the test1 and test2 sets of the challenge get F1 scores of 51.0% and 51.4%, respectively. The F1 scores are poor even though the visualizations indicate good prediction outcomes. They, therefore, compare the prediction findings to the currently available annotations and identify some inconsistencies. As a result, they also offer suggestions on ways to enhance the tagging of this dataset.

Figure 14. Example segmentation of Detectron 2. Source: (Y. Wu et al., 2019)

2.10.3 You Only Look Once (YOLO)

The model's compact size and quick calculation speed form the basis of the YOLO target identification technique. The organization of YOLO is simple. The neural network can immediately output the position and category of the bounding box. Because YOLO simply has to upload an image to the network to receive the final detection result, it can quickly complete time-based video detection. YOLO employs the global image directly for detection, which may encrypt the information at the global level and lessen the error of mistaking the background for the item. Because YOLO can acquire highly generalized features that may be applied to many areas, it has a good generalization capacity (Jiang et al., 2022; Redmon et al., 2016). Target detection becomes a regression problem as a result, although detection precision still has to be increased. The test results for objects in groups and at close range are subpar for YOLO. Only two of the grid's boxes are anticipated, and they only belong to a new class of objects within the same category, which results in an atypical aspect ratio, in addition to other factors like a limited capacity for generalization (Jiang et al., 2022).

The positioning error is the primary factor in improving the detection efficiency because of the loss function. It is necessary to improve the handling of both large and small objects. Finding a loss function that strikes the right balance between these three factors is key to successful implementation. To increase the effectiveness of the detection, YOLO employs several lower sampling layers and does not use all of the target properties that were learned from the network (Jiang et al., 2022). Twenty-four convolution layers precede two fully coupled layers in the original YOLO design. Non-maxima suppression is the practice of selecting the bounding boxes with the highest Intersection Over Union (IOU) with the ground truth out of several bounding boxes predicted by YOLO for each grid cell (Jiang et al., 2022; Redmon et al., 2016).

There are numerous studies conducted within fisheries that utilize the YOLO algorithm. The study conducted by J. Li et al., 2022, proposed an upgraded CME-YOLOv5 network to detect fish in dense groups and small targets to lessen environmental disturbances and to address the issues of many fish, dense, mutual occlusion, and challenging identification of small targets. On the other hand, Wageeh et al., 2021 introduced a technique that enhances fish detection and fish trajectories in tough water conditions. The initial step in enhancing confusing photographs is to employ an image enhancement algorithm. The improved photos are then subjected to an object detection algorithm to find fish. The coordinates of the identified items are ultimately used to extract features like fish count and trajectories. By expanding the detection scale from 3 to 4, applying k-means clustering to increase the anchor boxes, developing a unique transfer learning technique, and enhancing the loss function, Raza & Hong, 2020 suggested an enhanced YOLOv3. They used the YOLOv3 architecture to conduct object detection on four fish species-specific datasets. They achieved mean average precision of 87.56%. (mAP). The mAP increased from 87.17% to 91.30 when we compared the experimental analysis of the old YOLOv3 model with the improved one. Many other domains utilize the YOLO algorithm. This study will only focus on all domains encompassing precision fishing and aquaculture.

Figure 15. YOLO Inference on Synthetic Cups Images

2.11 SYNTHESIS

Precision fishing involves the use of advanced technologies to enhance the efficiency of commercial fishing operations. It incorporates the use of computer vision, a branch of artificial intelligence, to aid in identifying fish and other marine life, and the generation of synthetic image data to train these algorithms and improve their accuracy.

Computer vision's role in precision fishing is to analyze images and video feeds from underwater cameras to identify fish and other marine species. This contributes to more targeted fishing operations, aiding in species preservation and preventing overfishing. A significant challenge in implementing computer vision for precision fishing is the limited availability of high-quality training data. The accuracy of these algorithms can be significantly improved by generating large volumes of synthetic image data.

Synthetic data, created using computer graphics software simulating realistic underwater environments and marine life movements, offers several advantages. It allows the creation of extensive training data essential for enhancing computer vision algorithm accuracy. It provides highly controlled and varied data, which boosts the algorithm's performance under diverse conditions. Moreover, it is a faster and more cost-effective solution compared to real-world data collection, which can be time-consuming and challenging.

This thesis's central focus is the development of an AI web service, facilitating the broader application of these technologies. This web-based platform democratizes the use of advanced technologies, potentially leading to their wider adoption and resulting in substantial benefits for the fishing industry and marine ecosystems. By making the service accessible online, it can reach a broad audience, including those who might not have had access to such technologies before.

However, deploying and maintaining such a service isn't without its challenges. These could include ensuring the robustness of the service, dealing with a high volume of data and users, and updating the underlying models as new data and advancements in AI and computer vision emerge. Despite these challenges, the benefits of such a service are significant, paving the way for future work in this domain.

In conclusion, precision fishing is a critical application of computer vision and synthetic image data generation. With the development of accessible web-based platforms, we can expect further advances in precision fishing, leading to a more sustainable and efficient fishing industry.

Chapter 3

DESIGN AND DEVELOPMENT METHODOLOGY

This chapter describes several data collection and analysis strategies that were used that are pertinent to the study. The methodology will cover things like the research design, methods of data gathering, requirement specification, software architecture, analysis and design, datasets, model design, implementation, and evaluation and inference metrics. Additionally, the researchers reviewed the data case analysis and ethical considerations associated with conducting the study in this chapter.

3.1 RESEARCH DESIGN

This study employed two research methodologies to interpret the data and results collected to meet the specified objectives within the research.

The first methodology was a quantitative research method with an experimental research design. This was used to develop and evaluate the deep learning algorithms and computer vision models, which were the core components of the web application. The performance metrics of these models were statistically analyzed to infer their efficacy and reliability.

The second methodology was a qualitative, theoretically grounded, descriptive research design. This involved conducting structured interviews, grounded in the ISO 25010 standards. The goal was to assess and interpret stakeholders' opinions and sentiments regarding the implementation of the web application in the fisheries and aquaculture industry. This methodology centered on characterizing the system's qualities, users' experiences, and potential impact on the fisheries and aquaculture industry, rather than explaining why specific events occurred.

The central focus of this study, titled 'Aquaria: A Web-Based AI Platform Utilizing Convolutional Neural Networks and YOLOv8 for Precision Fishing, Aquaculture, and Marine Sustainability in the Philippines', was the development and assessment of the Aquaria web application. This application leveraged advanced computer vision algorithms and AI technologies to enhance precision fishing and aquaculture practices in the Philippines.

The evaluation of the Aquaria application was designed around the ISO 25010 software quality model, assessing the system's functionality, efficiency, reliability, usability, maintainability, and portability, among other factors. This structured evaluation process ensured a thorough and comprehensive appraisal of the developed application, allowing for identification of potential areas for improvement.

Data collection for this research was conducted through structured interviews, designed to gather insights, opinions, and feedback about Aquaria from potential users and industry experts. This qualitative user feedback played a critical role in the iterative process of development, refinement, and evaluation of the Aquaria web application.

The research design ensured a rigorous, user-centric, and iterative approach to the development, refinement, and evaluation of the Aquaria application. The goal was to provide an innovative, valuable, and effective tool for the local fishing and aquaculture industry in the Philippines.

3.2 METHODS OF DATA GATHERING

The methods of data gathering will vary according to each specific use case. In terms of collecting the data for generating synthetic images that will be used to train and validate the deep learning models, the researcher will leverage the use of existing images available on the web or publicly shared in online image repositories. One image repository that will be utilized is Flickr. Digital media like images and videos can be uploaded, arranged, and shared via the media platform Flickr. To efficiently scrape the images that will be used in this study, the Flickr API will be utilized using the Python programming language. Other images will also be taken from free stock images websites like Unsplash, Pexels, and Pixabay.

On the other hand, for the qualitative assessment of the features and implementation of the web application, the primary data will be gathered through an online survey with respondents varying across different fields encompassing data science, machine learning, artificial intelligence, fisheries and aquaculture, and other relevant fields.

3.2.1 Synthetic Image Generation using Cut, Paste, and Learn Algorithm

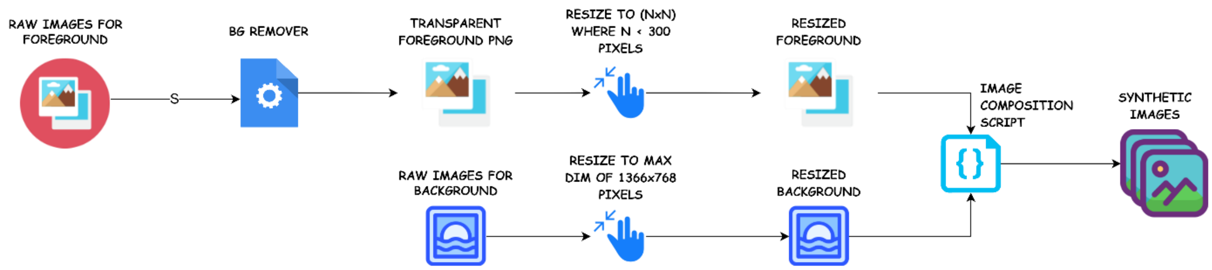


Figure 16. Synthetic Image Data Generation Flowchart

To generate the synthetic images using the Cut, Paste, and Learn Algorithm, raw images of foreground and background images will be collected throughout different image repositories available for online public use. The raw foreground images will first be subjected to a manual background removal process through the *remove.bg* website. Successful inputs will be downloaded on the local machine and the unsuccessful ones are disregarded. Then, both the transparent foregrounds and raw backgrounds will be subjected to a resizing process to scale all dimensions within the same range. This is also necessary to compress the file size and ensure that the images do not fully overlap the background when generating masks. The foreground images will be scaled to a NxN matrix where N is a pixel value less than 300 (N < 300px). On the other hand, the background images will also be scaled to a maximum dimension of 1366x768 pixels. After the resizing process, the images will now be subjected as inputs to the image composition Python script. Table 2 outlines the necessary parameters to run the *image_composition.py* script in the CLI.

Table 2. *Parameters for image_composition.py script*

Parameter	Description
input_dir	The directory containing the processed foreground and background images
output_dir	The directory to store the synthetically generated images together with the masks and corresponding JSON files
count	Number of synthetic images to be generated
width	Width of the synthetic image to be generated
height	Height of the synthetic image to be generated

Additionally, the image composition script will generate two (2) JSON files containing the mask definitions and dataset information. These files will be used as input to another python to provide the corresponding COCO annotations for the created synthetic images that will be used when training the model using the Computer Vision algorithms. Table 3 outlines the necessary parameters to run *coco_json_utils.py*.

Table 3. *Parameters for coco_json_utils.py*

Parameter	Description
md	JSON file containing the mask definitions generated by running image_composition.py
di	JSON file containing the dataset information generated by running image_composition.py

Alternatively, a web application will also be deployed to perform the same operations but without the necessary use of the CLI Python commands. The raw foreground and background images will just be uploaded on the website and upon clicking the "Generate" button, the synthetic images will be generated and downloaded together with the JSON files and COCO annotations. The parameters could also be set using different input parameters present within the website through sliders and text inputs with the scripts running on the backend.

3.2.2 Research Instruments for Synthetic Image Data Generation

In the synthetic image data generation part of the study, Python version 3.x is necessary to utilize the scripts and import the dependencies needed for the algorithm to run efficiently and effectively. Table 4 outlines the dependency requirements for the *image_composition.py* while table 5 outlines the dependency requirements for the *coco_json_utils.py* script.

Table 4. *Dependency requirements for image_composition.py script*

Package	Version Requirement	Implementation
JSON	2.0.x	import json
Warnings	Python built-in	import warnings
Random	Python built-in	import random

NumPy	1.23.2 or above	import numpy as np
DateTime	Python built-in	from datetime import datetime
Pathlib	1.0 or above	from pathlib import Path
tqdm	4.64.x or above	from tqdm import tqdm
Python Image Library	9.2.x or above	from PIL import Image, ImageEnhance

Table 5. *Dependency requirements for coco_json_utils.py*

Package	**Version Requirement**	**Implementation**
JSON	2.0.x	import json
Scikit-Image	0.19.x	from skimage import measure, io
Shapely	1.8.x	from shapely.geometry import Polygon, MultiPolygon
Numpy	1.23.2 or above	import numpy as np
Python Image Library	9.2.x or above	from PIL import Image, ImageEnhance
Pathlib	1.0 or above	from pathlib import Path
tqdm	4.64.x or above	from tqdm import tqdm

Any Integrated Development Environment (IDE) can be used if it has Python support and optionally a built-in terminal to run the scripts. When the user opts to utilize the CLI when generating the synthetic images, the scripts should be run within the python folder given that the scripts are also stored in that directory. Otherwise, it will not work.

As discussed in section 3.2.1, the image_composition.py script will be taking an input and output directory. These directories should be called as per the specified directory of the data_synthetic folder. The placement of the processed foreground and background images is important, especially for clustering the images within proper class types. The supercategory folder can be renamed to any general class to be trained, while the category folder refers to the specific class. The name of the category folder will determine the class names the images under that directory will belong to during the training and validation phase. The output directory set in the output directory parameter of the script should correspond to the location as structured within the folder. Therefore, it is important to label this accordingly as it will be used to segregate train, validation, and testing images. Finally, the masks folder will contain the masks images of the generated synthetic images. The number of masks should be the same as the number of the generated image and the order should also coincide. Otherwise, wrong inputs will be introduced to the training stage of the deep learning process.

3.2.3 Data Gathering for Qualitative Assessment

In the context of this study, qualitative data gathering was an essential process that was in alignment with the ontological, epistemological, and axiological justifications of the research. Given the aim and objectives of the study, the ISO 25010 standards served as the guide for structured interviews. These interviews were the primary method for qualitative data collection, providing valuable insights on the practicality and efficacy of the 'Aquaria' application in the context of precision fishing and aquaculture in the Philippines.

In terms of participant selection, a purposive sampling technique was used, as it most closely aligned with the study's goals. Purposive sampling is a method for identifying and selecting cases or participants that are most likely to produce the relevant and useful information, making efficient use of limited research resources (Campbell et al., 2020; Palinkas et al., 2015).

The justifications for using a purposive sampling strategy are based on the idea that certain types of individuals might possess unique and valuable perspectives on the theories and issues being investigated. For this study, stakeholders, including fishermen, fish market vendors, aquaculture farmers, and others who interact with the entities present in the 'Aquaria' application, were selected for their distinctive insights and experiences.

The structured interviews, grounded in the ISO 25010 standards, facilitated a deep understanding of the application's performance, usability, and relevance in the context of the participants' work. The data collected through these interviews contributed significantly to the qualitative assessment of the application, offering a user-centric perspective that guided the iterative refinement and evaluation of 'Aquaria'.

Through purposive sampling and ISO 25010-guided interviews, the research obtained data from those who fit the study objectives most closely. This approach ensured that the results were relevant to the research context, providing a rich source of insights for understanding and enhancing the 'Aquaria' application's impact on precision fishing and aquaculture in the Philippines.

3.2.4 Research Instruments for Qualitative Assessment

The research instrument that will be used in this study is an online questionnaire administered via Microsoft Forms to collect data from the target respondents. This questionnaire will contain inputs that will determine the feasibility of the web applications as well as their implications for the fisheries industry. The framework of the questionnaire was established by the researchers' readings, preliminary studies, relevant literature, and applicable published and unpublished theses. The conditions required for creating an effective data collection tool were considered when creating the instrument.

The questionnaire will be divided into three parts. The first part of the questionnaire provides an overview and rationale of the study being conducted as well as its intended purpose. The structure will follow the Problem-Agitate-Solution-Benefit (PASB) format to better help the respondents gain a gist of the overall study as well as the benefits brought by its objectives to the target community. The second part will be utilized to take data about the respondent's information like age, sex, demographic, source of income, and occupation within the fisheries sector or related field. The results gathered in this section will be utilized for data analysis using different statistical concepts within data science. There will also be a sample demonstration of the web application and a link to the landing page. Furthermore, the respondents will also be required to answer questions regarding the different aspects and components of the application to meet the Software Development Quality Standard. The third part of the research will contain text inputs asking about the opinions, preferences, and subjective outlook of the target respondents about the web application. The results in this step will be necessary for clustering the respondents based on sentiments using sentiment analysis.

To help respondents prepare their knowledge, a comment indicating the subject was dimmed will be placed as a disclaimer to elicit confidentiality of the responses. To elicit flexibly formatted viewpoints on the subjects or issues, open-ended questions were employed. The preference for the structured questionnaire is based on several research presumptions, such as the fact that it is the least expensive way to gather data, prevents personal bias, is less persuasive for an immediate response, and permits respondents to stay anonymous.

3.2.5 Open Source and External Data Acquisition

A significant portion of the data utilized in the development of Aquaria services was gathered from open source datasets and the contributions of other researchers in the field. This diversified and enriched our training data, augmenting the capabilities of our models and enabling them to handle a wide variety of marine life, marine debris, and market marine animals.

The open-source datasets and external data were carefully vetted for relevance to our research and model objectives. The datasets were then meticulously preprocessed, standardized, and integrated into our existing data infrastructure to ensure consistency and optimal model performance. We appreciate the open nature of the data science community, and express our gratitude to all the authors and contributors of these datasets.

These additional data sources were instrumental in training the 'Deep Fish', 'Marine Debris', 'Fishing Vessels', 'Marine Animals', 'Market Marine Animals', and 'TACO' services of Aquaria. The variety and depth of these datasets contributed significantly to the robustness and accuracy of these services.

We have endeavored to maintain academic integrity by appropriately citing and referencing these data sources throughout our research paper. The open source and external datasets used were integral to the success of our research and greatly broadened the scope of the Aquaria services.

3.3 THE DATASETS

3.3.1 Custom Fish Vision Dataset

A controlled environment aquarium of 1 meter by 0.5 meters (1m x 0.5m) could be an ideal setup for a computer vision inference task involving the observation of fish behavior. With four cameras positioned on the sides of the aquarium and lighting on the corners, it would be possible to capture high-quality images and videos of the fish from multiple angles.

To ensure that the fish can swim and behave naturally, it would be important to create a suitable habitat within the aquarium. This could include adding decorations such as plants and rocks to provide hiding places and visual interest, as well as ensuring that the water is properly filtered and oxygenated.

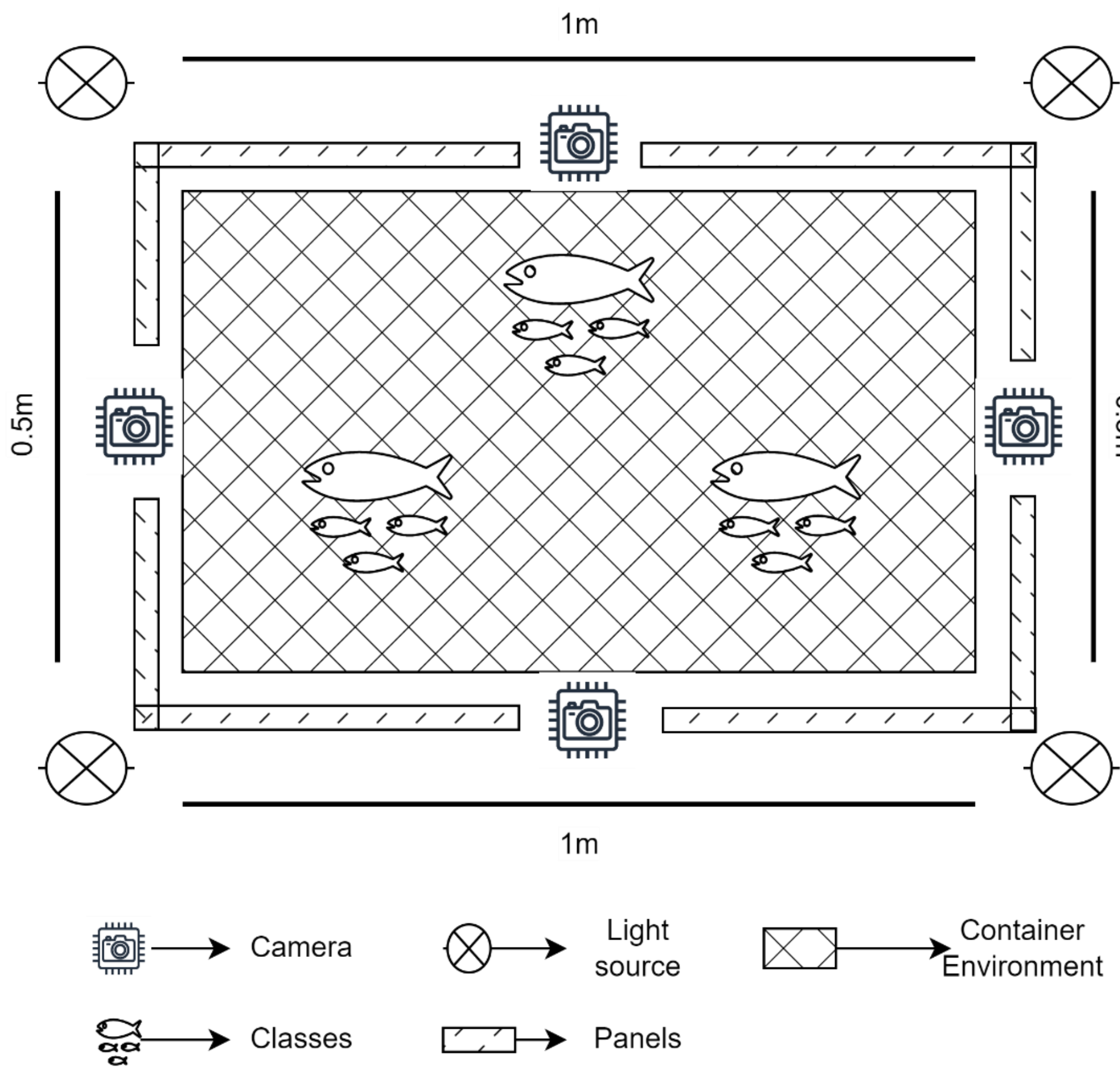


Figure 17. Custom Environment Schematic Diagram

To ensure a consistent background in every image captured, the aquarium could be surrounded by panels that maintain a uniform appearance. This would help to eliminate the influence of external factors on the images and allow for a more accurate analysis of the fish's behavior.

In terms of the fish themselves, a variety of four species with three fish per category would provide a good balance of diversity and consistency. This would allow for the observation of differences in behavior between the different species, as well as within each species. It would also be important to choose fish that are compatible with each other and the size of the aquarium, to ensure their well-being and the success of the study.

Figure 18. Materials Used in Building the Custom Environment

To gather images of the fishes, each individual fish classes were placed into the small aquarium to isolate them from the others and gather images within that small environment. The fishes were simultaneously placed after the other to curate a largscale fish dataset for the first web service in this study. The materials used for this custom environment can be seen on Figure 18.

The first web service developed in this study is titled "Custom Fish Vision Object Detection". In this service, a distinctive and substantial dataset was generated for four different fish species: Catfish, Perch, Tarpon, and Tilapia.

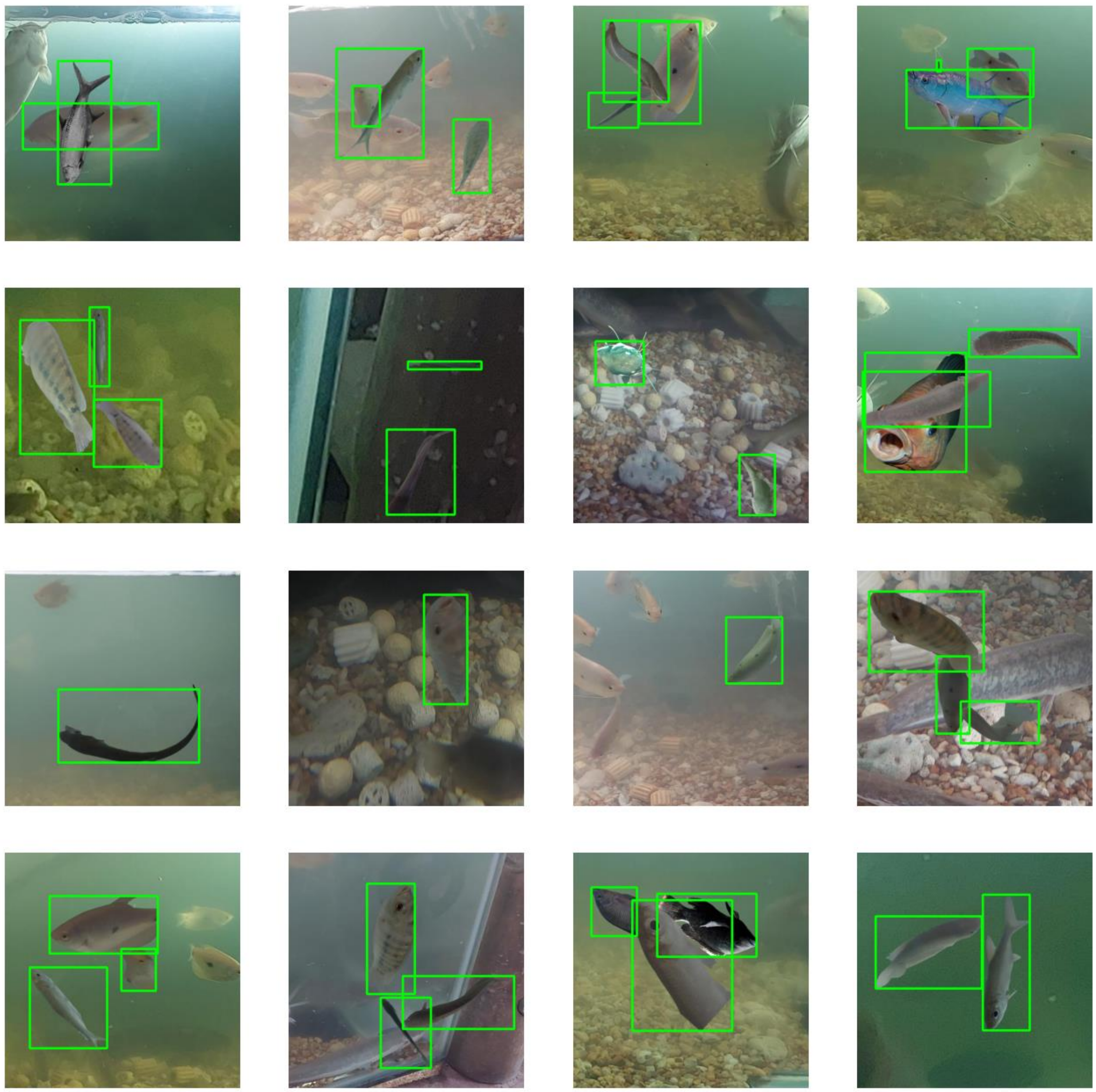


Figure 19. Sample Images from Custom Fish Vision Dataset with Annotations

The generation process involved procuring foreground images of the four fish species and applying the Cut, Paste, and Learn Image Synthesis technique. This technique, known for its effectiveness in generating high-quality synthetic images, helped in expanding the dataset size to a total of 30,000 images. By providing a wide range of sample data for each species, the technique enabled the development of a model capable of recognizing various conditions and positions of each fish type. The amassed images were then subjected to a training regimen using the state-of-the-art YOLOv8 Object Detection algorithm. This deep learning algorithm was employed to create an object detection model that could precisely and efficiently identify the four fish species in images.

3.3.2 Deep Fish Dataset

Figure 20. Sample Images from Deep Fish Dataset with Annotations

In the context of this study, the Deep Fish Service utilized the dataset from Saleh et al., 2020 as the foundation for developing and evaluating the deep learning algorithms and computer vision models that form the core of the Aquaria web application. By leveraging this comprehensive and meticulously curated dataset, the study ensured a rigorous evaluation and refinement process for the AI technologies integrated into the web application. The utilization of this dataset contributed significantly to the robustness and reliability of the deep learning models and the overall performance of the Aquaria web application in enhancing precision fishing and aquaculture practices in the Philippines.

3.3.3 Marine Debris Dataset

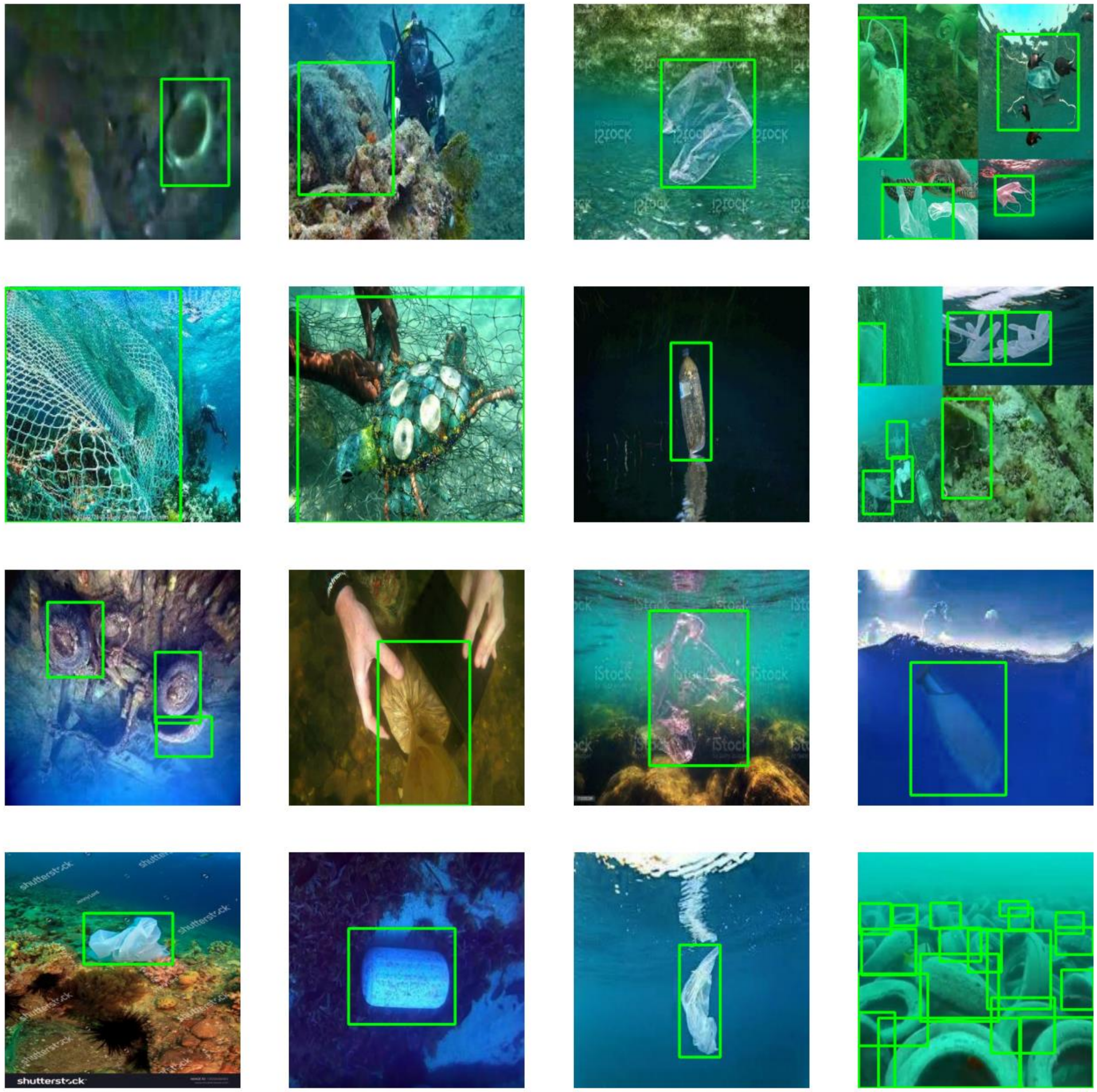


Figure 21. Sample Images from Marine Debris Dataset with Annotations

In this study, the Neural Ocean dataset was used to train and validate the computer vision models integrated within the Aquaria web application. These models were designed to detect and classify marine debris in underwater images, providing users with vital information about the prevalence and types of waste in specific areas. This functionality is particularly important for precision fishing and aquaculture operations, where the presence of marine debris can significantly affect productivity and sustainability.

3.3.4 Marine Garbage Dataset

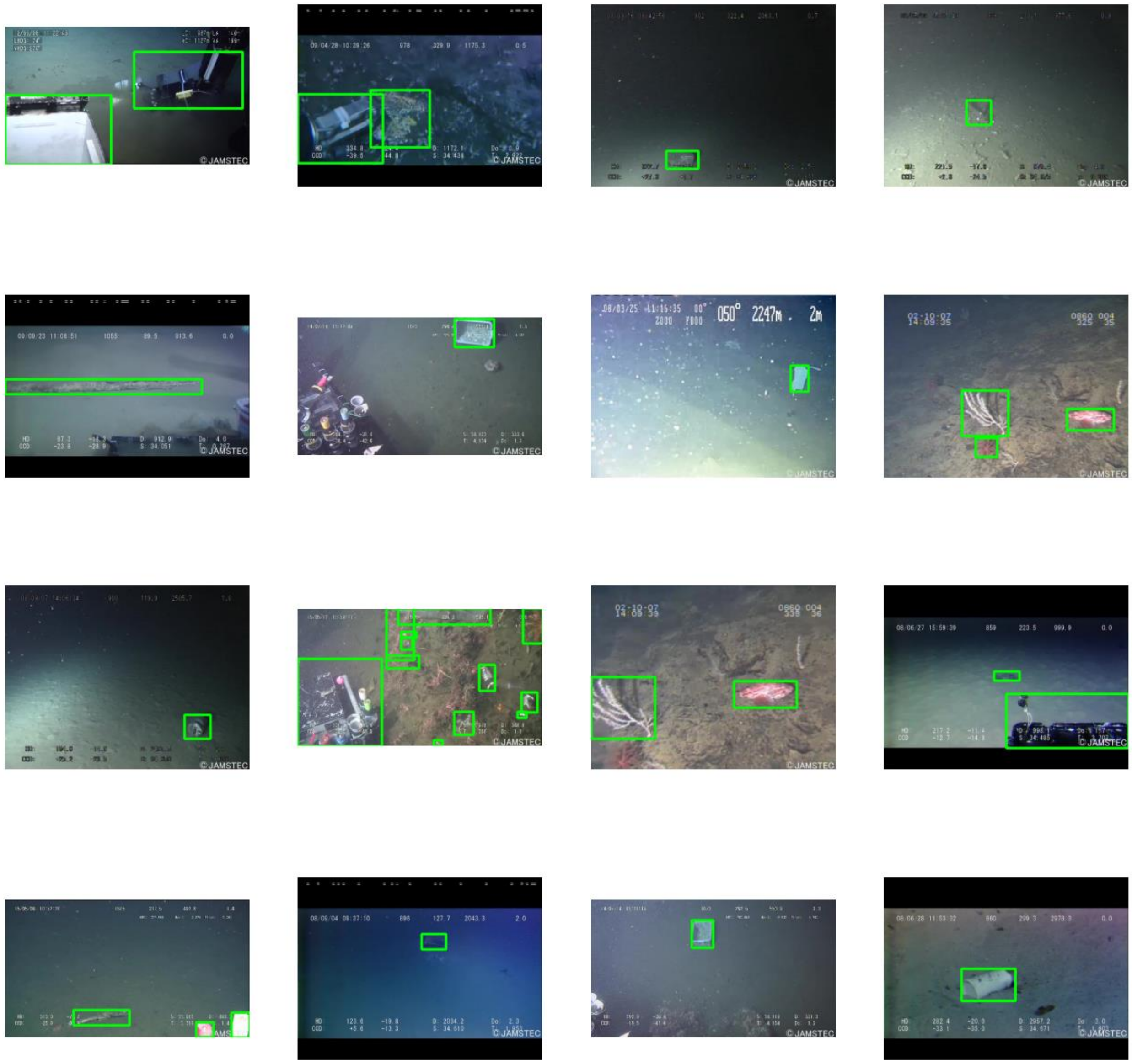


Figure 22. Sample Images from Marine Garbage Dataset with Annotations

In this study, the TrashCan dataset was employed to augment the computer vision models within the Aquaria web application. This dataset, created by Hong et al., 2020, features annotated images with instance segmentation annotations. These models were optimized to detect and classify marine garbage in underwater images, based on the rich, instance-segmentation annotations provided in the dataset. The valuable insights gained from this dataset significantly enhanced the Aquaria application's capabilities, contributing to its overall effectiveness and efficiency in promoting sustainable practices in the fisheries and aquaculture industry.

3.3.5 Fishing Vessels Dataset

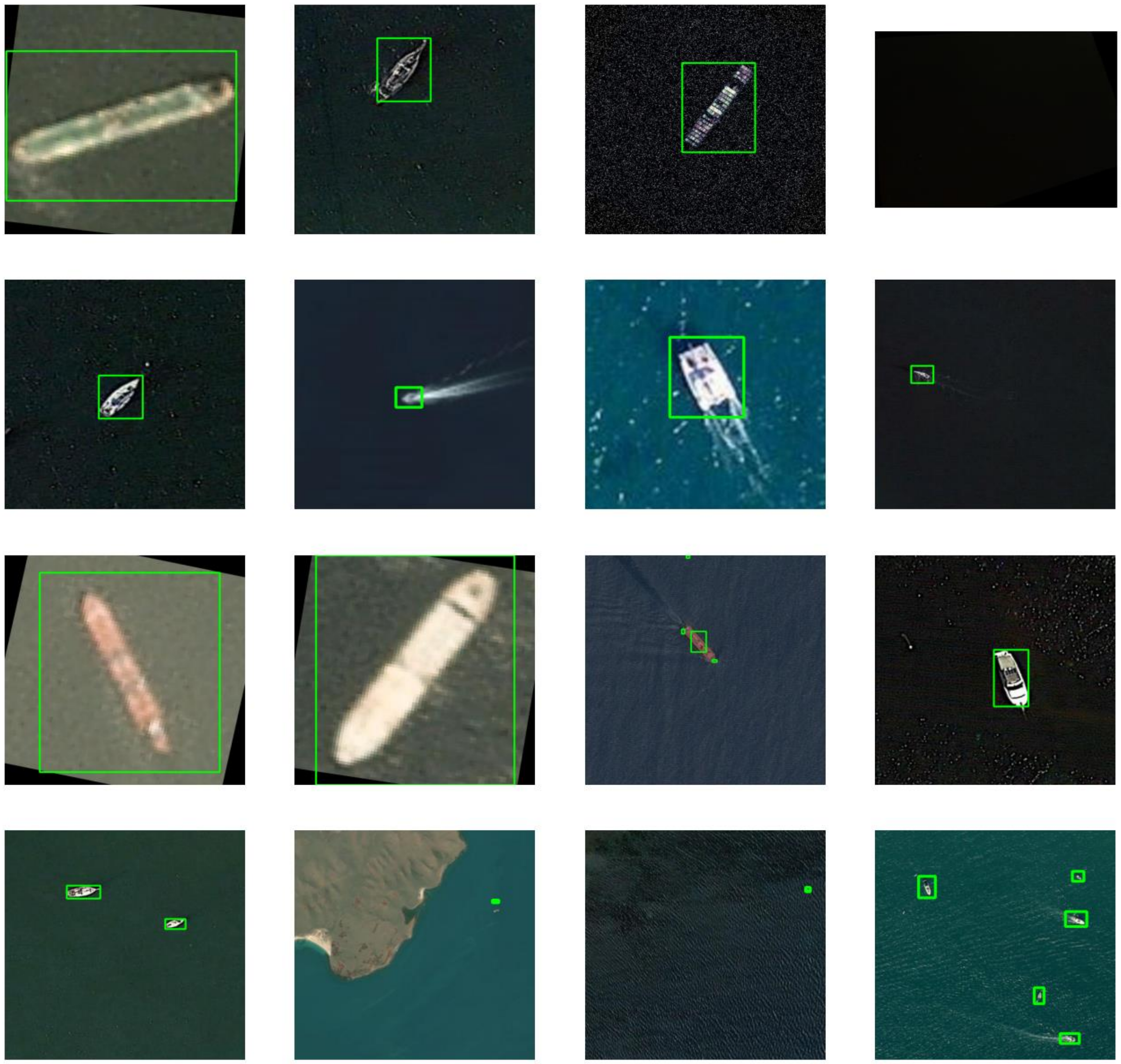


Figure 23. Sample Images from Fishing Vessels Dataset with Annotations

The fishing vessels dataset used in the research was sourced from a competition hosted on Hugging Face Spaces titled "Ship Detection." This dataset, rich in the variety of ship types and perspectives it provided, was instrumental in the development of the deep learning algorithms used within the Aquaria web application. Its specific focus on fishing vessels was particularly pertinent for the detection and identification of various fishing activities, thereby significantly improving the efficacy of the Aquaria application in monitoring and managing sustainable fishing practices in the Philippines.

3.3.6 Marine Animals Dataset

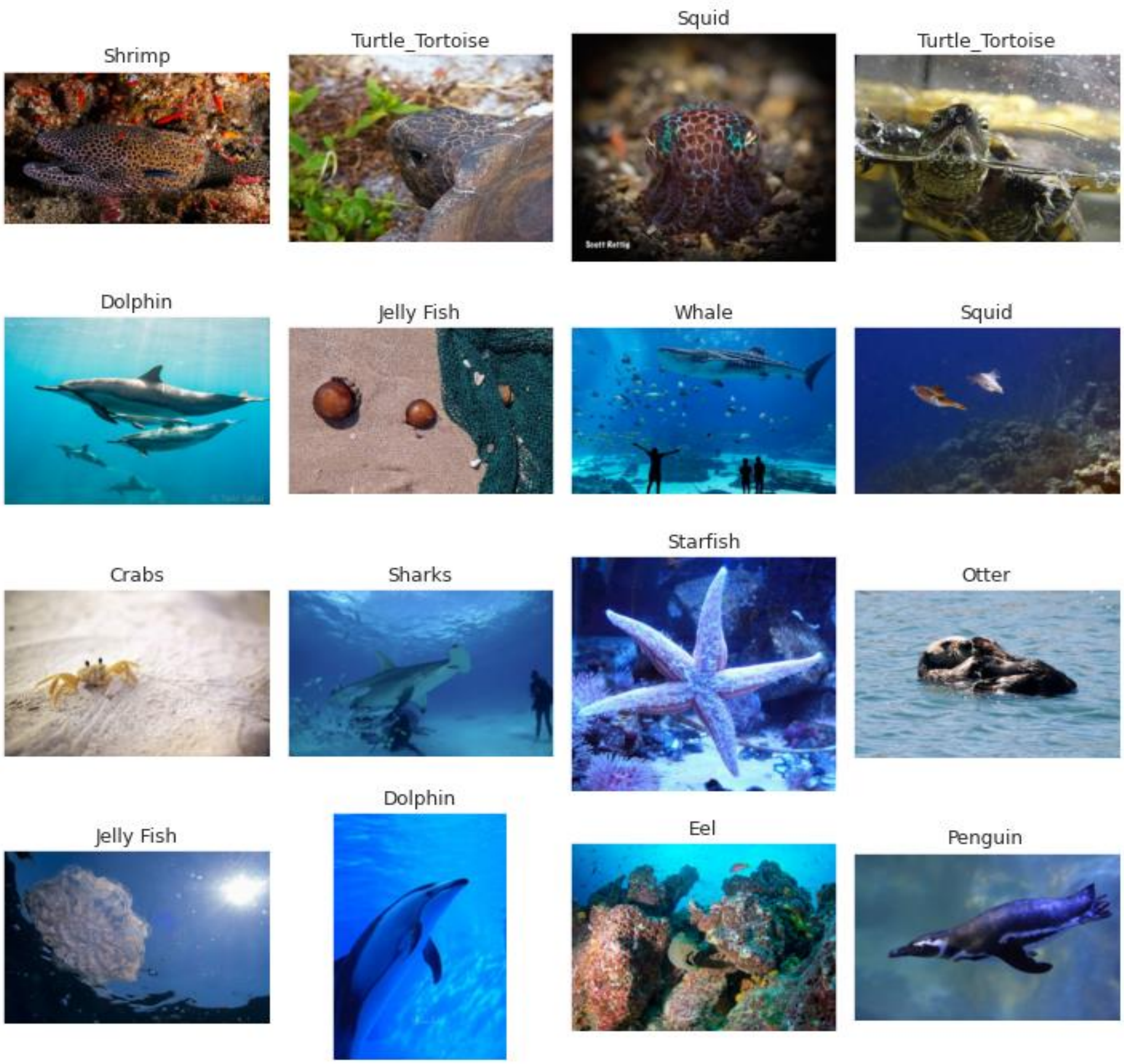


Figure 24. Sample Images from Marine Animals Dataset with class labels

The marine animals dataset was a unique compilation of images created by the researcher for the purposes of this study. A variety of marine animal images were collected from different sources such as Pixabay and Flickr, with the appropriate licenses and attributions considered. Pixabay images, free of licenses or attribution requirements, were combined with images from Flickr, where commercial use required crediting the original authors. The dataset, as of the study's duration, comprised 23 distinct classes, though it was designed with the potential for future expansion. All images were resized to ensure compatibility and efficiency with the applied machine learning algorithms, adhering to a standard of either (300px, n) or (n, 300px), where 'n' is a pixel size less than 300px. This personally curated dataset provided a tailored and extensive resource for training and testing the deep learning algorithms incorporated into the Aquaria web application.

3.3.7 Market Marine Animals Dataset

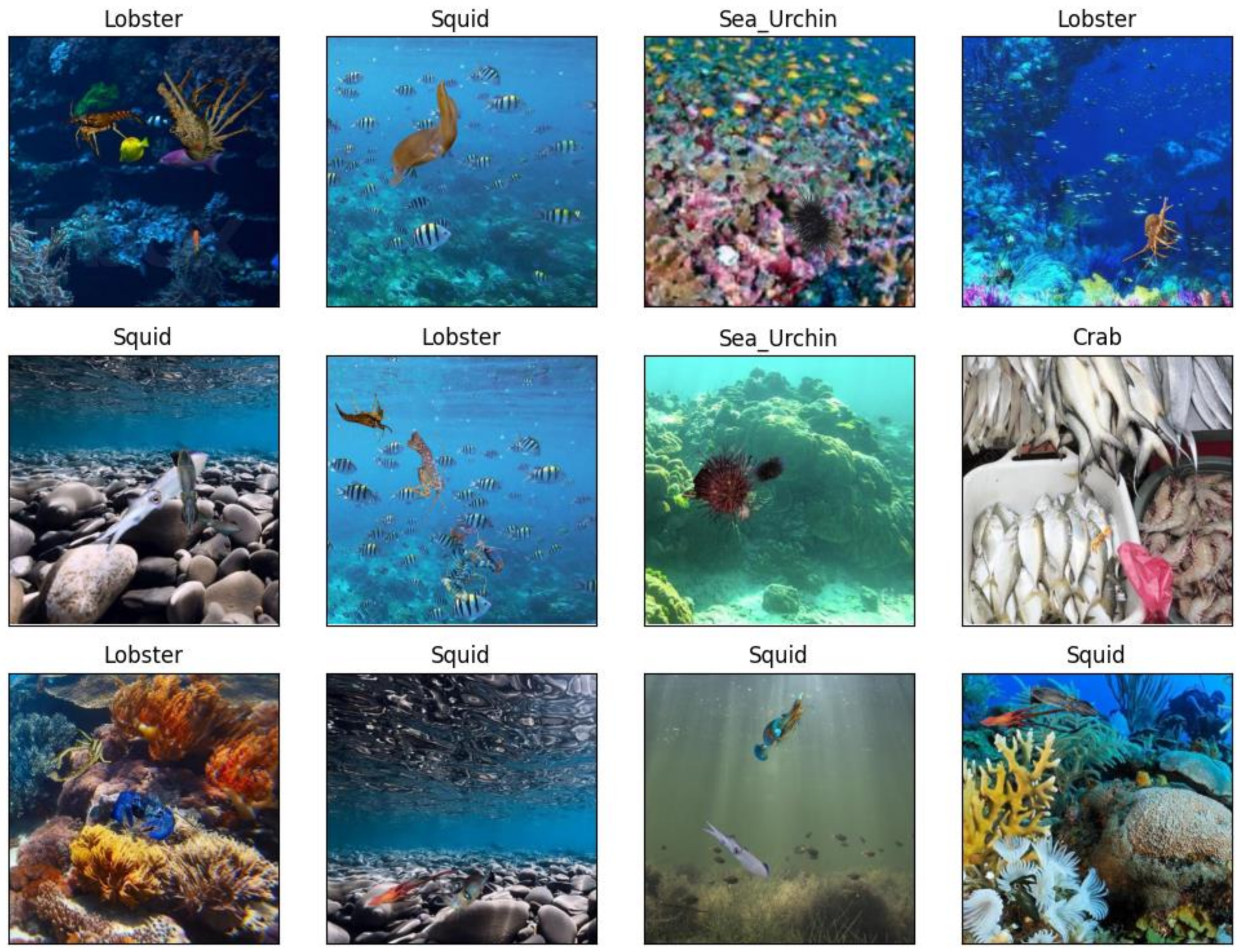


Figure 25. Sample Images from Market Marine Animals Dataset with class labels

The market marine animals dataset, featuring four classes of marine animals - squid, lobster, crab, and sea urchin, was assembled through a procedure comparable to that used for the custom fish vision dataset. Primarily, the dataset relied on synthetic images created by the Cut, Paste and Learn (CPL) algorithm. The CPL algorithm uses a simple yet effective method to generate new, realistic images by cutting, transforming, and pasting objects from a source image into a new context. By leveraging this approach, a diverse and rich set of synthetic images was created that significantly enhanced the ability to train the machine learning models used in the study. This artificial data augmentation strategy allowed for a comprehensive and robust dataset that effectively supported the Aquaria web application in recognizing and classifying these specific marine species.

3.3.8 Trash Annotations in Context (TACO) Dataset

Figure 26. Sample Images from TACO Dataset with Annotations

The TACO (Trash Annotations in Context) dataset was another essential dataset used in this research. Available at tacodataset.org, it is a comprehensive dataset aiming to facilitate litter detection in the environment. This dataset includes images of various categories of litter annotated using bounding boxes. The images in the TACO dataset are diverse and span multiple environments, from city streets and rural areas to beaches and oceans, thereby providing a vast range of data points that enhance the model's ability to recognize and classify litter in different environments and contexts. The use of this dataset improved the ability of the Aquaria web application to accurately detect and classify marine debris, contributing to the study's main objective of promoting cleaner oceans and a sustainable fishing industry.

3.4 ETHICAL CONSIDERATIONS

3.4.1 Ethical Considerations about the Application of Computer Vision in Precision Fishing

Precision fishing also referred to as intelligent fishing or smart fishing, is a fishing method that uses computer vision to effectively locate and catch fish more. By lowering the quantity of unintentionally caught fish and increasing the quality of the catch, this method has the potential to significantly enhance the sustainability of the fishing industry. But using computer vision for precision fishing also brings up some moral questions that need to be carefully thought out.

The risk of overfishing is one ethical issue with precision fishing. It might be feasible to detect and catch fish more efficiently by employing computer vision, which would allow for a catch rate that is higher than the ecosystem can support. The environment and the overall health of the ocean ecosystem may suffer as a result of the potential decline in fish populations. Precision fishing with computer vision will need to be closely monitored and regulated to reduce this danger and make sure that the ecosystem is not overfished.

The potential effect on fishermen's means of subsistence is another ethical issue with precision fishing. Fishing jobs could disappear if some tasks are automated as a result of the use of computer vision in the industry. Small-scale fishermen who depend on fishing for their livelihoods may suffer the most from this. The use of computer vision in precision fishing must be done in a way that supports fishermen's livelihoods and prevents job losses to address this problem.

The potential effects on fish health and welfare are a third ethical issue with precision fishing. The overhandling of fish that could result from computer vision in fishing could stress and harm the fish. Additionally, this might lead to a catch of lower quality and a decrease in the fish's market value. To solve this problem, it will be crucial to make sure that computer vision is used in precision fishing in a way that reduces stress and harm to fish.

In conclusion, the use of computer vision in precision fishing has the potential to greatly improve the sustainability of the fishing industry. However, it also raises some ethical considerations that must be carefully considered. These include the potential for overfishing, the impact on the livelihoods of fishermen, and the impact on the health and well-being of fish. To ensure that the use of computer vision in precision fishing is ethical, it will be important to carefully monitor and regulate its use and to take steps to mitigate the potential negative effects on the environment, fishermen, and fish.

3.4.2 Ethical Considerations of the Qualitative Data Collection

The set of rules that direct your study designs and procedures when conducting research are known as ethical considerations. When gathering data from people, scientists and researchers must always abide by a set of ethical principles. The integrity of science, respect for human rights and dignity, and cooperation between science and society all depend on research ethics. This study is conducted while upholding the ethical considerations necessary to make the information collected from the respondents and the derived results to be accurate and free from misconduct. The utmost priority of taking ethical actions and decision-making when conducting the study will not only benefit the respondents and target beneficiaries of the study, but also the researchers in understanding and valuing the integrity of the results produced and the process of attaining those results to help the fisheries industry. Table 6 outlines the ethical considerations together with the descriptions pertinent to their application within the study.

Table 6. *Ethical Considerations of the Qualitative Data Collection*

Ethical Consideration	Description
Voluntary Participation	Respondents will not be forced to participate in the study. They are free to opt-in or out of the study at any point in time and may choose to demand the removal of the data they have provided without any negative repercussions.
Informed Consent	Before they decide whether to participate, respondents will be made aware of the study's goals, advantages, risks, funding, and implications to the target beneficiaries.
Anonymity	The respondents' true identities won't be revealed. Personal

	information won't be gathered. Identifying information will be made available as optional input for identifying the responses of the respondents in the case that they would like to withdraw their answers.
Confidentiality	Although the researchers will not be aware of who the respondents are, the information gathered from them will still be kept a secret from others. To prevent others from connecting personally identifiable information upon inferring the data, the researchers will also anonymize the relevant information about the identity of the respondents. The data gathered will be stored in a secure repository accessed only by the researchers through several multi-factor authentications.
Potential for harm	The researchers will ensure that, as much as possible, the respondents will be free from harm in all aspects physically, socially, or psychologically. No living entity will also be harmed throughout the conduct of the study, especially during the data-gathering stage.
Result communication	The researchers will ensure the work is free of plagiarism or research misconduct and accurately represent the results inferred from the collected data.

3.5 SOFTWARE DEVELOPMENT METHODS

The primary method of agile software development the researchers will utilize in this study is Kanban. Kanban is a prominent lean methodology used in software engineering that is being more widely embraced by software companies. Because of its growing popularity, researchers are paying more attention to this issue (dos Santos et al., 2018). It necessitates complete transparency of work and real-time capacity communication. Team members can always observe the status of every piece of work thanks to the visual representation of work items on a kanban board.

The researchers chose Kanban as a method of software development due to the familiarity of the processes and comfort to do tasks based on different status segregation to see which works are urgent and backlogged. Several task cards were used each having its respective assignee and checklists. Since the researcher works alone, the sole researcher is the assignee to all the task cards. This also includes all the monitoring, checklists, and setting of the definition of done (DOD) of each task card.

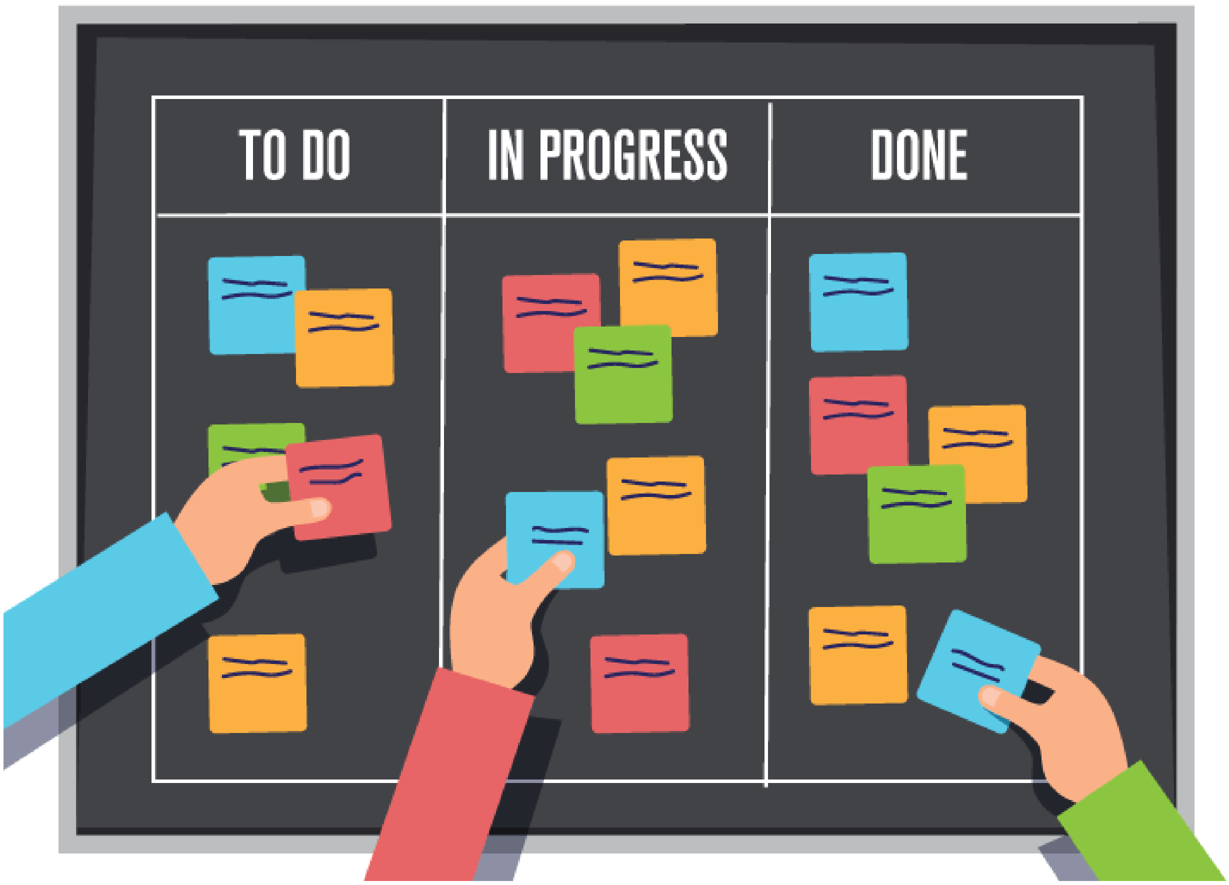


Figure 27. Kanban Board Example

3.6 REQUIREMENT SPECIFICATION

3.6.1 Project In-Scope

The project was centered on the development of multiple web-based services utilizing computer vision to aid various aspects of marine science and industry. These services were developed and tested utilizing a wide array of data sources, each meticulously gathered, analyzed, and integrated to suit specific purposes.

The first service, the Fish Vision service, harnessed the capabilities of computer vision in identifying and classifying different species of fish. The source of data for this service was an adaptation from the realistic fish-habitat dataset presented by Alzayat et al., the Marine Debris dataset from Roboflow, and the TrashCan dataset published by Hong, Fulton, and Sattar. These datasets provided a rich source of annotated images, featuring diverse marine life, underwater trash, and debris, which facilitated precise and robust fish and debris detection algorithms.

Another service developed was the Fishing Vessel Identification service, which utilized a dataset from a competition hosted on the Hugging Face Spaces platform, intended for ship detection. This service aimed to identify and classify different fishing vessels, a tool valuable for marine traffic monitoring and fisheries management.

Moreover, two services were created focused on marine animal identification. One utilized a custom-created dataset featuring images of marine animals from various online sources, while the other, the Market Marine Animals service, used synthetic images generated through the Cut, Paste, and Learn algorithm, focusing on four classes: squid, lobster, crab, and sea urchin. These services aimed to foster a better understanding of marine biodiversity and aid in the monitoring of seafood market dynamics.

Lastly, the TACO Marine Debris service was designed to recognize different types of marine debris. For this, the project utilized the TACO dataset, a rich source of litter images in different environments. This service was crucial for promoting efficient marine debris management strategies.

In summary, this project's scope was concentrated on the development and implementation of multiple, purpose-specific computer vision services integrated into a web application, designed to address various needs in marine science and industry. The project, through meticulous data sourcing, algorithm development, and web service implementation, aimed to promote efficient and sustainable practices in these sectors.

3.6.2 Project Out-Scope

While the scope of the project was comprehensive and versatile, certain elements were deemed outside its purview.

The first was the collection of original data. Although various datasets were used in the development of the services, the process of gathering new, original data through fieldwork or via direct means, such as deploying cameras or drones to capture new images or videos, was outside the project scope.

Secondly, the project was focused on the development of web-based services and did not extend to hardware design or implementation. Building or customizing physical devices, such as underwater cameras, drones, or autonomous vehicles for data collection, was beyond the project's reach.

Thirdly, the deployment and maintenance of the web-based services in a production environment were not covered within the project. While the services were developed and thoroughly tested, considerations such as scalability, real-time performance under production loads, and long-term maintenance, including server uptime, troubleshooting, and user support, were beyond the scope.

Lastly, the project did not involve the training or education of end-users. Although the developed services were designed with user-friendliness and accessibility in mind, providing structured training programs or workshops for fishermen, marine biologists, or other potential users was not included within the scope of the project.

In summary, while the project's ambit was to create robust and versatile computer vision-based web services for diverse marine applications, aspects relating to original data collection, hardware development, production deployment, and end-user training fell outside its purview.

3.6.3 Hardware Requirements

The hardware requirements for training a computer vision algorithm for precision fishing will depend on the size and complexity of the algorithm, as well as the amount of training data available. In general, the hardware should be able to support the computational demands of training a machine learning model, including the ability to process large amounts of data quickly and efficiently.

One option is to use a powerful desktop or laptop computer with a fast CPU and a dedicated GPU. This type of hardware can support the training of complex algorithms and handle large amounts of data. Alternatively, the training can be done on a cluster of computers or a cloud-based platform, which can provide even more computing power and parallel processing capabilities.

Figure 28. Minimum Hardware Requirements

Another important consideration is the amount of memory available on the hardware. Machine learning algorithms often require large amounts of memory to store the training data and intermediate results. As such, the hardware should have enough memory to support the training process without running out of space.

Table 7. *Hardware Requirements*

Hardware	**Minimum Requirements**	**Recommendation**
Processor	Quad-core CPU	Octa-core CPU or higher
Memory	8 GB	16 GB or higher
Storage	256 GB SSD	512 GB SSD or higher

Graphics Card	NVIDIA GeForce GTX 1650 or equivalent	NVIDIA GeForce RTX 3070 or higher
Operating System	Windows 10, Linux (Ubuntu 16.04 or higher)	The latest version of Windows or Linux
Network	Gigabit Ethernet	10 Gigabit Ethernet or higher
Power Supply	500W	750 W or higher
Cooling	Air cooling	Liquid cooling

3.6.4 Software Requirements

In Table 8, we list the specific Python packages required for the implementation of the project. Each package's name, version, and a brief description of its functionality are presented. These packages include machine learning libraries such as PyTorch, TensorFlow, and PaddlePaddle, image processing tools like OpenCV and Albumentations, data processing and visualization libraries such as NumPy, Pandas, and Matplotlib, among others. They are crucial for the development, training, and deployment of the convolutional neural networks and YOLOv8 models used in the project.

Table 8. *Technical Software Dependencies*

Package Name	Version	Description
numpy	1.23.4	A fundamental package for array computing with Python.
albumentations	1.3.0	A fast and flexible image augmentation library.
boto3	1.26.100	Amazon Web Services SDK for Python.
colorama	0.4.6	Python library for colored terminal text and cursor positioning.
comet_ml	3.33.3	Machine learning platform for tracking, comparing, explaining and reproducing experiments.
coremltools	6.3.0	Tools for converting machine learning models to Core ML format.
filterpy	1.4.5	Library for Kalman filtering and optimal

		estimation.
Flask	2.2.3	A lightweight WSGI web application framework.
ipython	8.11.0	Command shell for interactive computing in multiple programming languages.
matplotlib	3.7.1	Library for creating visualizations in Python.
mss	9.0.1	Ultra-fast multiple screenshots module.
onnx, onnxruntime, onnxsim	1.14.0, 1.15.0, 0.4.28	Open Neural Network Exchange, a format to represent deep learning models.
openai	0.27.2	Python interface to the AI models developed by OpenAI.
opencv-python-headless	N/A	Open-source computer vision and machine learning software library.
openvino	2022.3.0	Comprehensive toolkit for developing applications that emulate human vision.
paddlepaddle	2.1.3	Open-source deep learning platform developed by Baidu.
pafy	0.5.5	Python library to download YouTube content and retrieve metadata.
pandas	1.5.3	Powerful data structures for data analysis, time series, and statistics.
Pillow	9.5.0	Python Imaging Library for image processing.
psutil	5.9.4	Cross-platform library for process and system monitoring in Python.
pycocotools	2.0.6	Libraries for the MS COCO dataset, with APIs for loading, parsing, and visualizing annotations.
PyYAML	6.0	YAML parser and emitter for Python.
Requests	2.31.0	A simple, yet elegant HTTP library.
scipy	1.10.1	Python-based ecosystem of open-source software for mathematics, science, and engineering.
seaborn	0.12.2	Python data visualization library based on

		matplotlib.
setuptools	67.6.0	Easily download, build, install, upgrade, and uninstall Python packages.
Shapely	2.0.1	Manipulation and analysis of geometric objects.
scikit-image	N/A	Collection of algorithms for image processing.
streamlit, streamlit_extras	1.20.0, 0.2.7	Framework for building and sharing data apps.
tensorflow, tensorflow_intel, tensorflowjs	N/A	Open source platform for machine learning developed by Google Brain Team.
tensorrt	8.6.1	High performance deep learning inference optimizer and runtime.
tflite_runtime, tflite_support	2.12.0, 0.4.3	Tools for running TensorFlow Lite models.
thop	0.1.1.post2209072238	Compute the FLOPs and MACs of PyTorch model.
torch, torchvision	2.0.0, 0.15.1	Scientific computing framework with wide support for machine learning algorithms.
tqdm	4.65.0	Fast, extensible progress bar for Python and CLI.
tritonclient	2.33.0	Triton Inference Server for deploying AI models at scale in production.
ultralytics	8.0.99	Particle physics and AI software research lab.
wget	3.2	Pure Python download utility.
x2paddle	1.4.1	Tool to convert models from other formats to PaddlePaddle format.

Table M outlines the necessary system packages used in the project. These include libraries for developing user interfaces (libgtk2.0-dev), tools for OpenGL (freeglut3-dev, libgl1-mesa-glx), and packages for optical character recognition (Tesseract-OCR and its related libraries). The system packages support the underlying computational and interface requirements of the project.

Table 9. *System Packages*

Package Name	Version	Description
freeglut3-dev	N/A	Free-software/open-source alternative to the OpenGL Utility Toolkit (GLUT) library.
libgtk2.0-dev	N/A	Toolkit for creating graphical user interfaces.
libgl1-mesa-glx	N/A	Mesa OpenGL library.
tesseract-ocr, libtesseract-dev, libtesseract4, tesseract-ocr-all	N/A	Optical character recognition engine for various operating systems.

3.6.5 Website Requirements

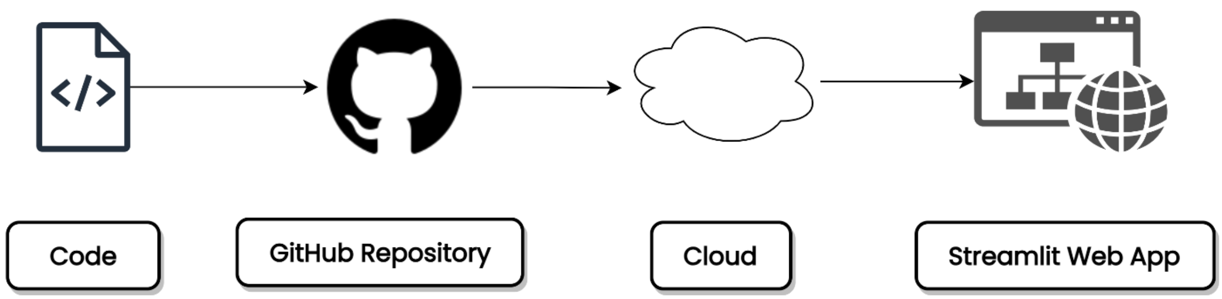


Figure 29. Streamlit CI/CD Flowchart

Table 10 outlined the requirements for the deployment of the Aquaria website. A GitHub account was needed, serving as the main repository for the website's code and other relevant files. The repository comprised a collection of these files and folders, including the code, content, and multiple assets that made up the website.

The machine learning models, which were essential for the computer vision tasks of the Aquaria website, were stored in an Amazon Web Services (AWS) S3 Bucket. Upon the initial loading of the webpage, these models were automatically retrieved from the S3 Bucket and saved to the local Streamlit Cloud repository. This procedure ensured that the models were readily accessible for immediate use, thereby improving the performance and user experience of the website.

In this study, Streamlit Sharing was selected as the web hosting provider, providing the necessary infrastructure and support for hosting the website on the internet. Lastly, a distinct domain name was required, which served as the web address that users keyed into their browser to access the Aquaria website.

Table 10. *Website Requirements*

Requirement	Description
GitHub account	The account is used to host website code and other files on GitHub
Repository	Collection of files and folders that make up a website, including code, content, and assets
Machine learning model	The algorithm used to make predictions or perform tasks on the website
Deployment service	Service that connects GitHub repository to a web hosting provider, and automatically deploys website when changes are made to the repository
Web hosting provider	Service that provides infrastructure and support for hosting websites on the internet
Domain name	The web address that users will enter in their web browser to access the website

3.7 ARCHITECTURE AND DESIGN

3.7.1 System Architecture

In the system architecture presented on Figure 30, the end-user, who could be any beneficiary of the Aquaria software, played an essential role. These users, utilizing either a mobile phone or a web browser from a personal computer, accessed the Aquaria website, which was hosted on Streamlit Sharing.

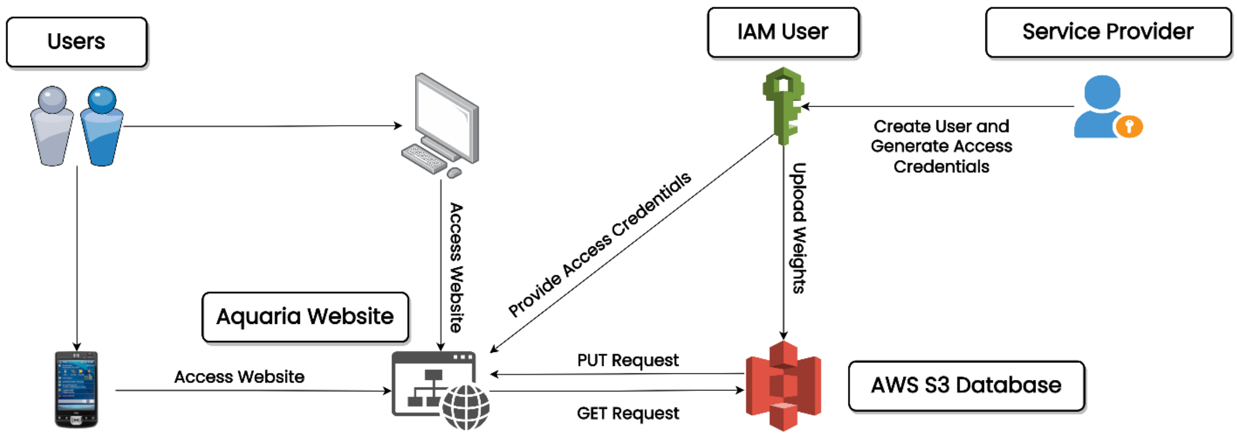


Figure 30. System Architecture

The Aquaria website had been granted capabilities to retrieve computer vision models from an Amazon Web Services (AWS) S3 bucket. The process for this retrieval involved using IAM (Identity and Access Management) access credentials, which were generated by the service provider or the root user. This ensured a secure transmission of models from the AWS S3 bucket to the Aquaria platform.

In this architecture, the AWS S3 bucket also held the capability to PUT models onto the local Streamlit Cloud repository where the Aquaria platform was deployed. This flexibility was essential for easy updates or changes to the models being used by Aquaria.

Moreover, the IAM user, furnished with the appropriate credentials, also had the capacity to upload models to the S3 bucket, providing an efficient method for model updates or replacements. This mechanism allowed for quick model iterations, facilitating continuous improvements and adaptations to the AI capabilities of Aquaria.

3.7.2 Software Architecture

The software architecture of Aquaria has been designed to offer a seamless and interactive user experience. Initially, the user gains access to the landing page of the Aquaria website, which serves as the entry point to the variety of services provided by the platform. Upon choosing a specific service, the corresponding service page springs into action by reaching out to the AWS S3 bucket.

This is a crucial step where the required computer vision models, stored securely in the S3 bucket, are fetched and placed in the local environment of the Streamlit cloud. This process is facilitated by an IAM user who is responsible for managing access to the S3 bucket. The IAM user is generated by the service provider, and they are equipped with the ability to upload models onto the S3 bucket if necessary.

Once the models are available in the Streamlit local cloud environment, they become accessible to the respective services on the front-end of the Aquaria platform. This allows the services to leverage these models in real-time to deliver the chosen functionality to the user. In essence, the software architecture has been designed to ensure quick, secure, and efficient data flow to provide users with a high-quality experience while using the platform.

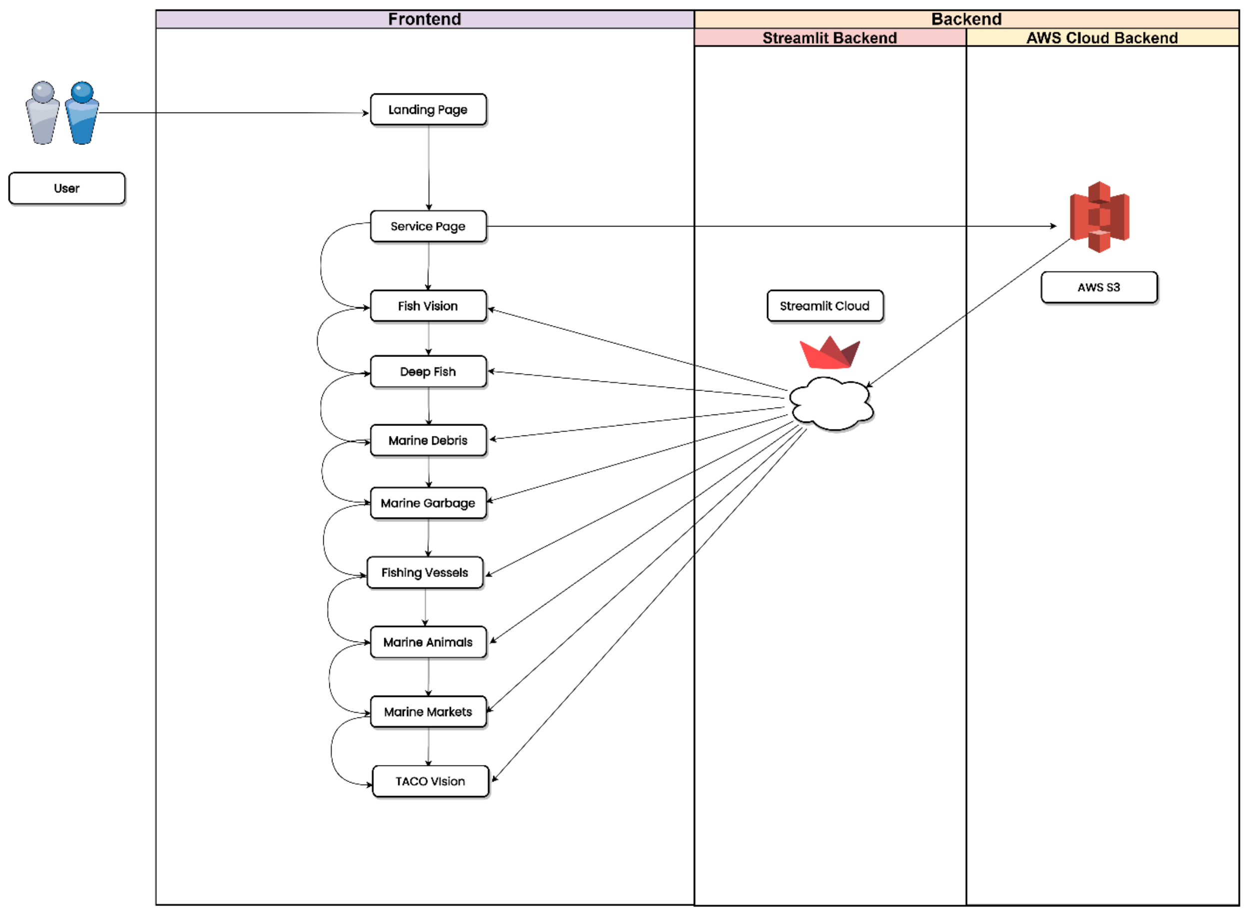


Figure 31. Software Architecture

3.8 UNIFIED MODELING LANGUAGE DIAGRAMS

3.8.1 Data Flow Diagram

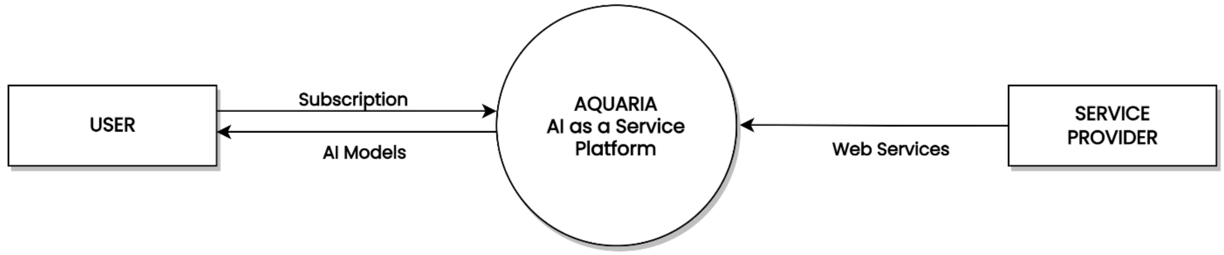


Figure 32. Level 0 Data Flow Diagram (Context Diagram)

Three distinct levels of Data Flow Diagrams (DFDs) were meticulously created to thoroughly elucidate the underlying flow of data within the Aquaria system. The Level 0 DFD, also known as the Context Diagram, provides a high-level overview of the entire system. This can be visualized in Figure 32. Expanding upon the context diagram, the Level 1 DFD, depicted in Figure 33, breaks down the system's main functions, providing a more detailed view of the system's interactions.

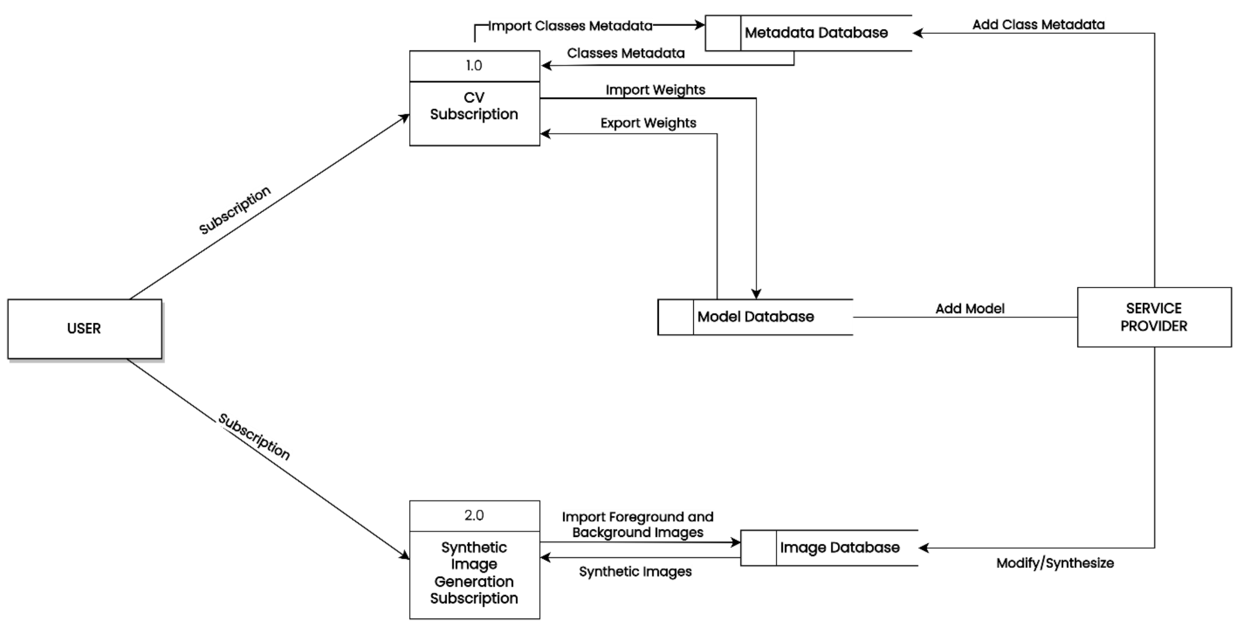


Figure 33. Level 1 Data Flow Diagram

Finally, the Level 2 DFD, represented in Figure 34, delves further into the intricacies of the system, detailing the subprocesses within the main functions illustrated in the Level 1 DFD. The progressive decomposition of the system via these three DFD levels furnishes a comprehensive understanding of the data flow within Aquaria.

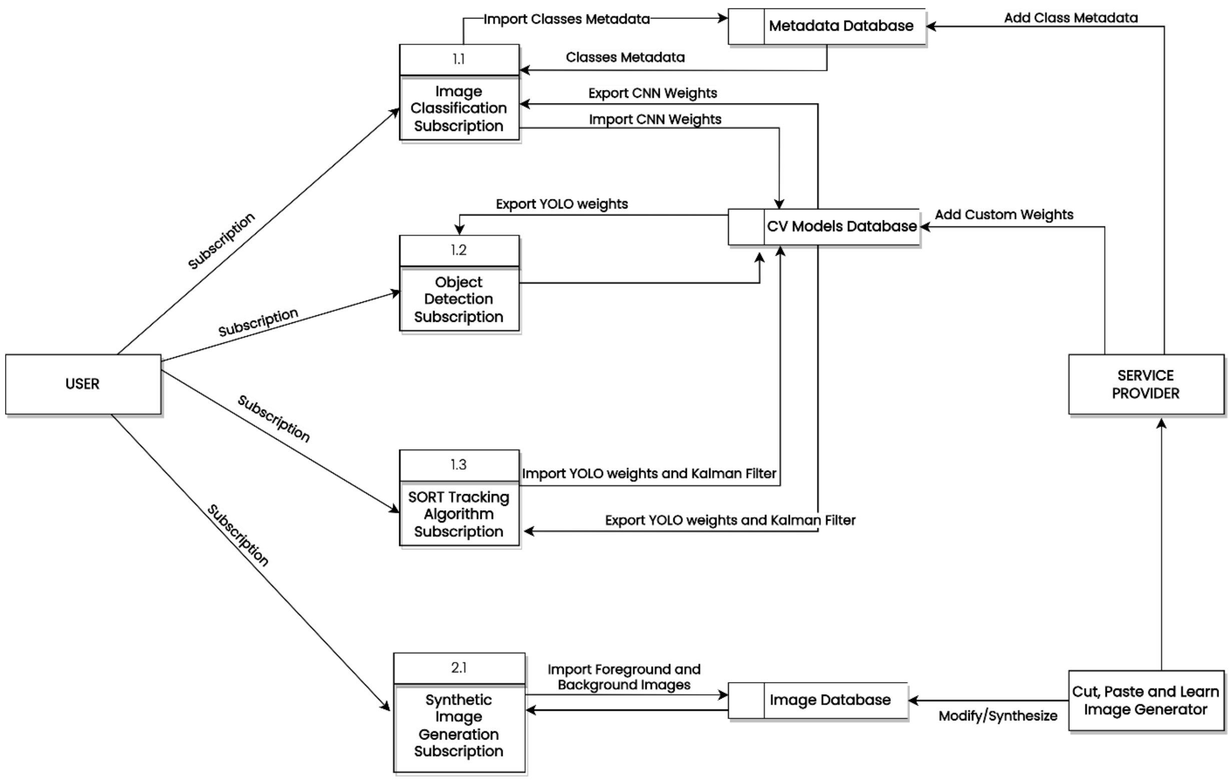


Figure 34. Level 2 Data Flow Diagram

3.8.2 Use Case Diagram

Two distinct use case diagrams were developed to elucidate the use cases for each service provided by Aquaria. Figure 35 illustrates the use case diagram specifically for the computer vision services, outlining the interactions between two primary actors: the user and the web services provider.

In this schema, the user actor was depicted as having the ability to access both the homepage and the AI service page, the latter being directly linked to the computer vision service page. This representation embodies the user's journey through the platform, from the initial point of entry to the specific utilization of the computer vision services.

On the other hand, the web services provider actor was represented with a broader set of capabilities. These included adding new computer vision services to the platform, accessing the model repository (i.e., the AWS S3 bucket), and accessing the computer vision services page. This element of the diagram serves to encapsulate the responsibilities and control the web services provider has over the services, models, and page access on the platform. The entirety of this diagram was designed to provide a comprehensive view of the interactions between the two main actors within the context of the computer vision services of the Aquaria platform.

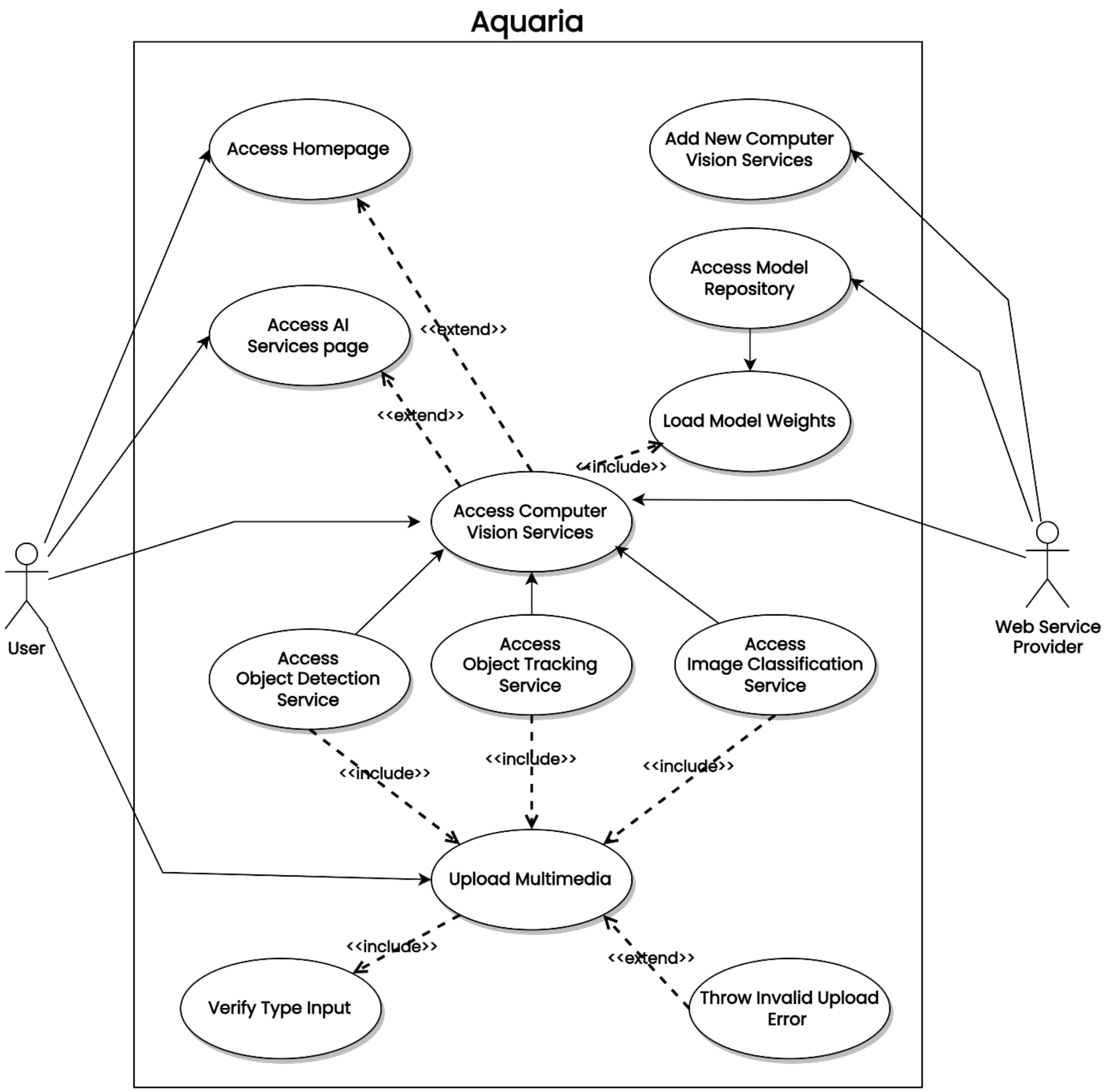


Figure 35. Use Case Diagram for Computer Vision Services

The second use case diagram, as depicted in Figure 36, was created to articulate the use cases pertaining specifically to synthetic image generation within the Aquaria platform. Unlike the computer vision services use case diagram, the synthetic image generation use case diagram is characterized by the presence of only one actor: the user.

In this diagram, the user actor was shown to have the capacity to access both the homepage and the AI service page. The AI service page leads directly to the Image Generation Service. Upon reaching this location, the user was granted the option to upload foreground and background images to the Synthetic Service for the generation of synthetic images.

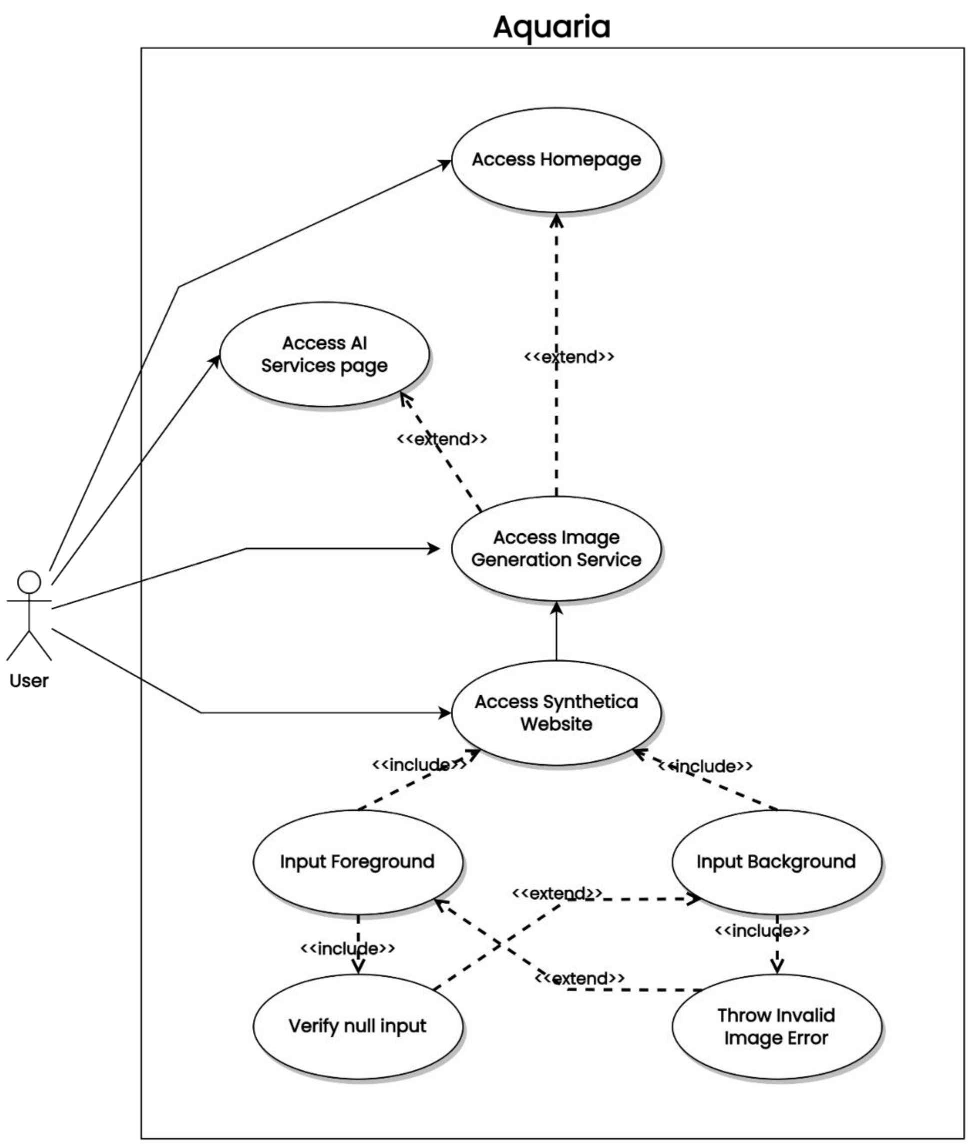


Figure 36. Use Case Diagram for Image Generation Services

This schematic was designed to encapsulate the interactions and pathways the user could navigate while seeking to generate synthetic images on the Aquaria platform. By detailing these actions and services, the diagram provided a comprehensive overview of the user's journey within the context of synthetic image generation.

3.8.3 Component Diagram

In Figure 37, the component diagram constructed for the Aquaria platform is displayed. This diagram aimed to provide a comprehensive view of the different components and their relationships within the system.

At the top level of the diagram, the user component is divided into two subcategories: Mobile User and Web User. Both of these user categories have been established to interface directly with the primary component, the Aquaria Website. This website component is subsequently split into two core sub-components: Computer Vision Services and Image Generation Services. The Aquaria Website also carries two key attributes: the Web UI, encapsulating the user-facing interface, and the Streamlit Integration and Hosting attribute which outlines the backend structure.

Each of the sub-components of the Aquaria Website hold their own unique attributes. For the Computer Vision Services sub-component, these include Object Detection Service, Image Classification Service, and optionally the Image Enhancement Service. Conversely, the Image Generation Service sub-component is denoted by a singular attribute: Synthetic Image Generation.

Crucially, the Computer Vision Services sub-component interfaces with the AWS Cloud component. Within this AWS Cloud component, the S3 Datastore, marked by its default persistence, exists as a sub-component. The S3 Datastore bears two attributes: Model Repository and API Access. These attributes are responsible for interfacing with the IAM User sub-component, also situated within the AWS Cloud component. This IAM User sub-component possesses three attributes: Permissions, Security Groups, and Access Credentials. Notably, access to the IAM User component was limited solely to the Service Provider component.

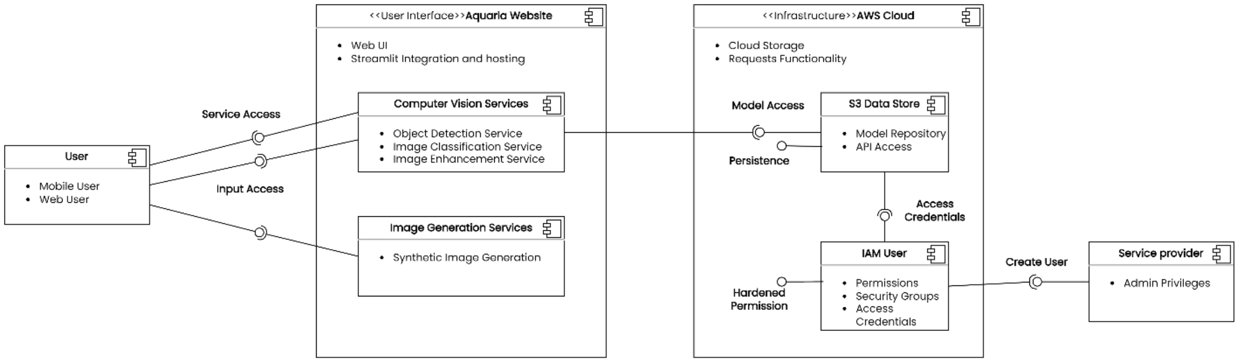


Figure 37. Component Diagram

3.8.4 Activity Diagram

Presented in Figure 38 is the general activity diagram of the Aquaria web service, illustrating the dynamic interactions among four major panels: User, Aquaria Web Service, AWS Cloud, and Service Provider. This diagram encapsulates a multitude of activities and flows, underlining the robust and comprehensive functionality of the system. Each panel represents an essential stakeholder or system component and their responsibilities within the entire process. It provides a comprehensive picture of the tasks, procedural steps, organizations or people involved, and the sequential flow of these elements in the entire operation. Detailed activities within each panel are well-delineated and arranged to depict the interactivity among these components, thereby illustrating the integral roles they play in the system's operation. This depiction aims to provide a high-level understanding of the system's activity flow, and it underscores the complexity and interconnectedness of the Aquaria web service's operation.

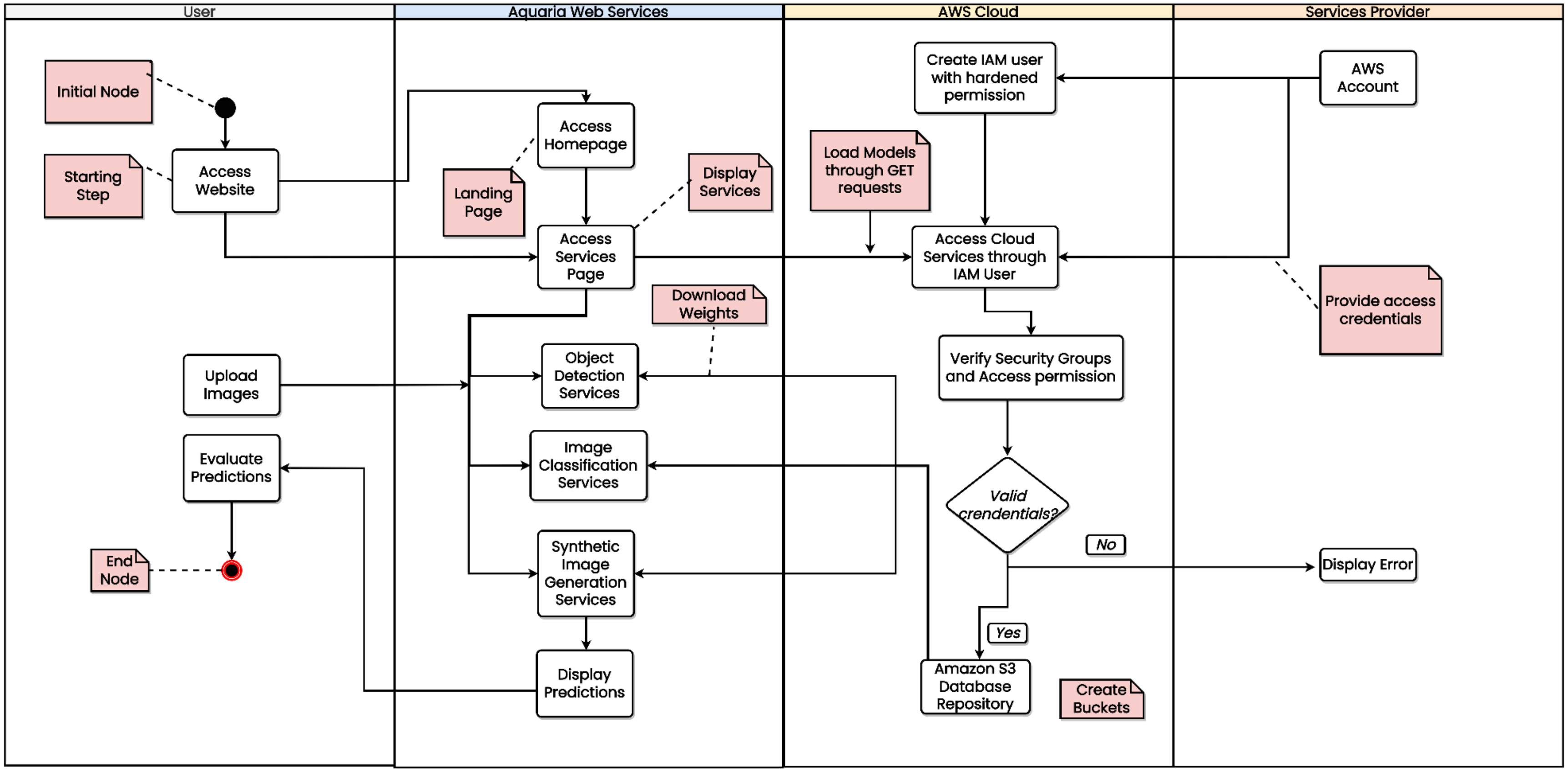


Figure 38. General Activity Diagram in Swim Lane Format

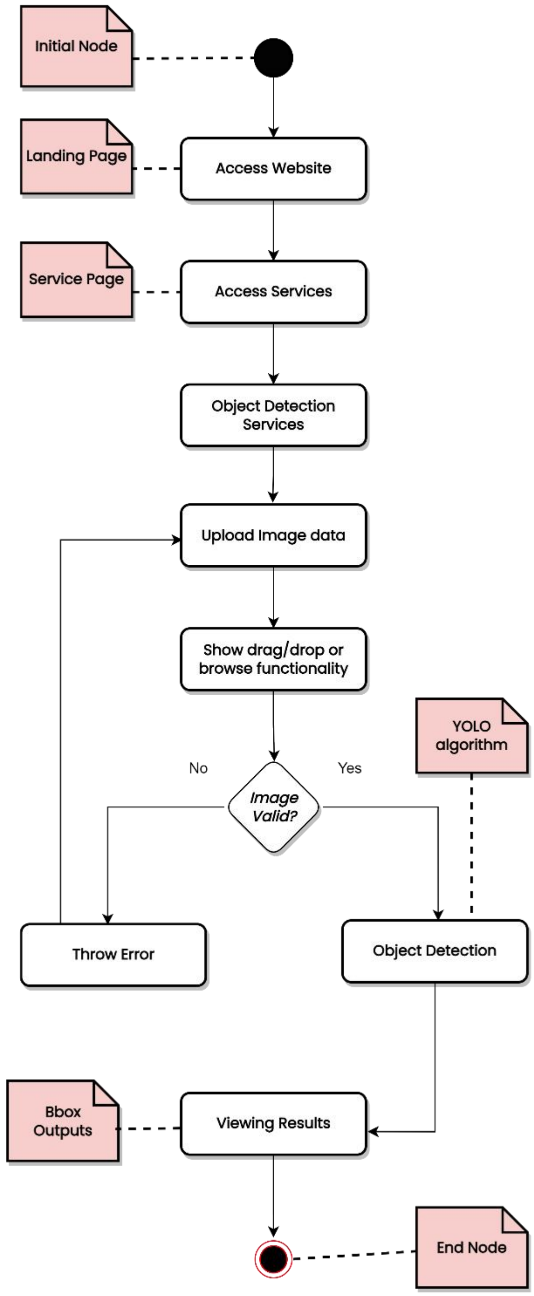


Figure 39. Activity Diagram for Object Detection Services

Figure 39 showcases the activity diagram specific to the Object Detection service offered by the Aquaria web platform. The process commences when a user accesses the Aquaria website, navigates through the landing page, and reaches the service page where multiple object detection services are available. Once the user selects a desired service, they are prompted to upload an image for analysis.

The subsequent step involves a validation check on the uploaded image. If the image fails this validation (i.e., if the image format or content is invalid or corrupt), the system will promptly generate an error message to inform the user of the issue. Conversely, if the image passes the validation check, it is then processed through the YOLOv8 inference system.

This robust system applies machine learning algorithms to the image, identifying and plotting bounding boxes around the regions where the model predicts the presence of specific classes or categories. The system effectively illuminates detected objects in the image, thereby providing the user with informative and precise object detection results.

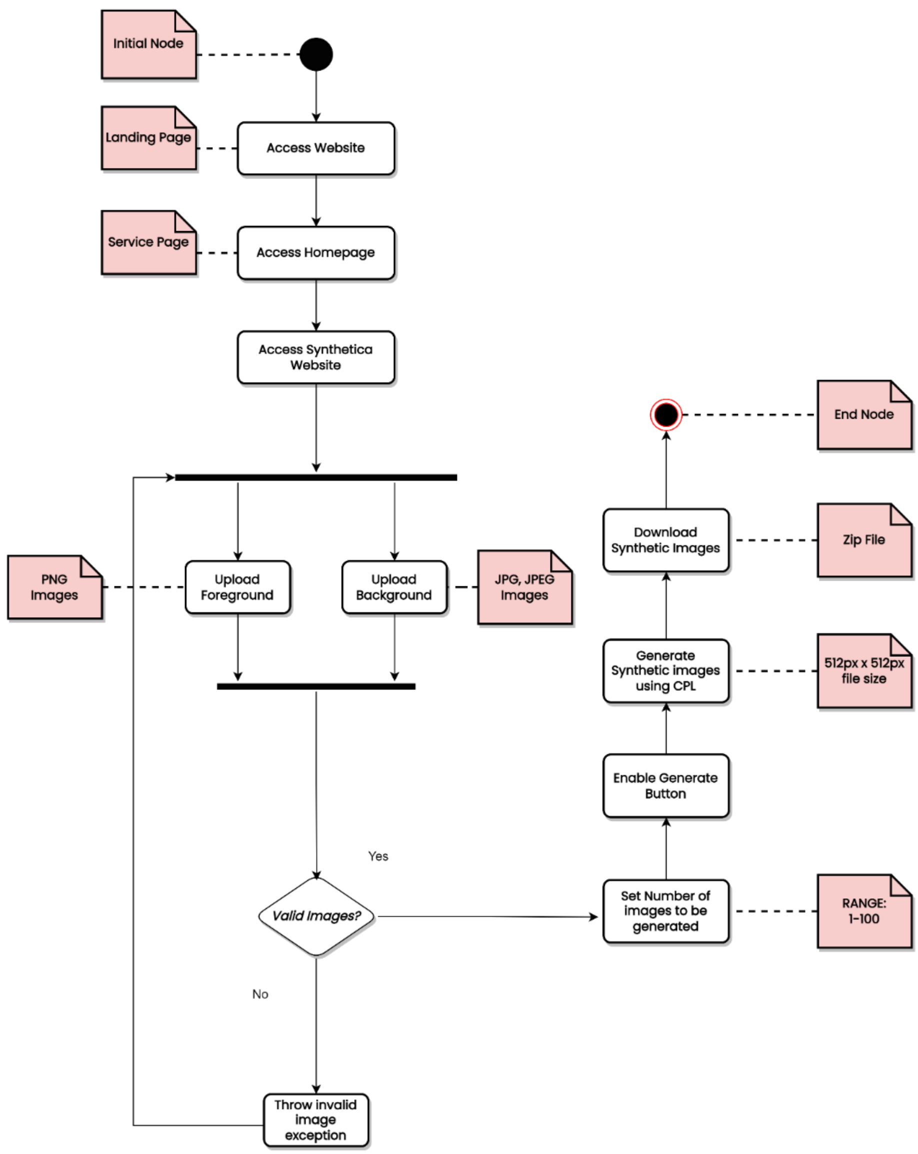


Figure 40. Activity Diagram for Synthetic Image Generation Service

Figure 40 presents the activity diagram specific to the Synthetic Image Generation service provided by the Aquaria web platform. Similar to the object detection service, the process commences when a user navigates through the Aquaria website's landing page, eventually landing on the service page. Here, they select the Synthetic Image Generation service.

Once the user has opted for this service, they are prompted to upload both a foreground and a background image. These uploaded images then undergo a backend validation process. If either or both images are found to be invalid, the system promptly throws an error to inform the user. However, if both images pass the validation checks, the user is then allowed to specify the number of synthetic images they wish to generate, ranging from 1 to 100.

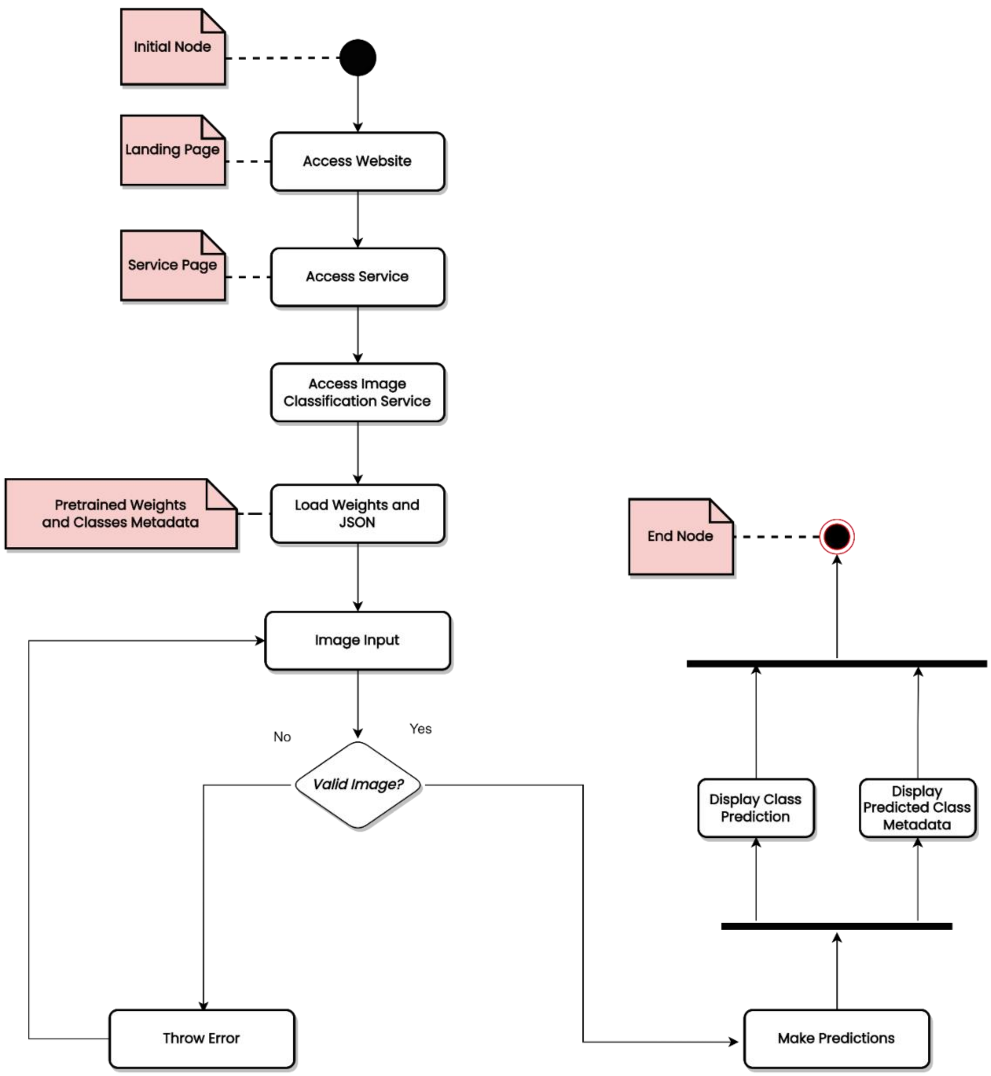


Figure 41. Activity Diagram for Image Classification Service

Upon specifying the desired quantity and pressing the 'Generate' button, the system initiates the synthetic image creation process. After all the synthetic images have been successfully generated, a 'Download' button is made available for the user. This allows them to download a ZIP file containing all the generated synthetic images, providing a user-friendly and efficient means of acquiring the output of this service.

Figure 41 illustrates the activity diagram specific to the Image Classification service provided by the Aquaria web platform. The user interaction flow largely mirrors that of the Object Detection service, beginning with the user accessing the Aquaria website, moving through the landing page, and then reaching the service page. Here, the user selects the Image Classification service.

The user is then prompted to upload an image, which is subsequently validated on the backend. If the image is found to be invalid, an error is thrown, alerting the user of the issue. However, if the image passes the validation checks, it is then processed by the image classification algorithm.

Unlike the Object Detection service, which produces bounding boxes around identified objects, the Image Classification service outputs the class to which the image belongs and displays corresponding metadata about that class. This provides users with a comprehensive understanding of the content of their image based on the classification determined by the service.

3.8.5 Sequence Diagram

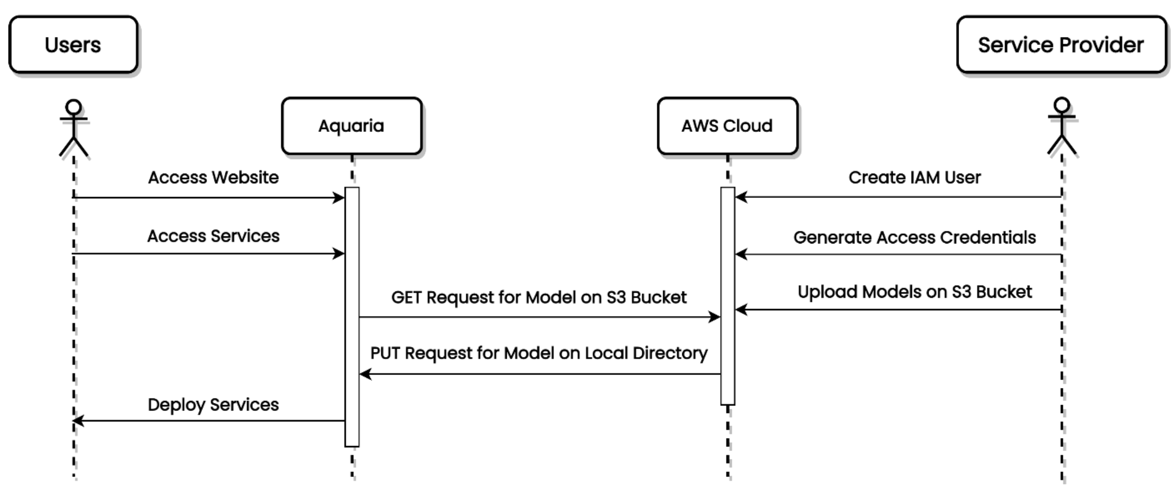


Figure 42. Sequence Diagram

Figure 42 presented the sequence diagram for the overall functionality of the Aquaria platform, illustrating the sequential interactions between the User, Aquaria Web Service, AWS Cloud, and the Service Provider. The sequence diagram captured the temporal sequence of the interactions and messages passed between the entities involved in the use of the Aquaria system.

Starting with the user initiating an interaction by accessing the Aquaria website, the diagram delineated the flow of activities as they proceeded across the platform. Following the user's request, the system responded by providing access to the required services and retrieving relevant resources from the AWS Cloud, as necessary.

The diagram also incorporated the role of the Service Provider, primarily involved in managing and providing access to the AWS resources, particularly in relation to the IAM User and the associated permissions, security groups, and access credentials.

3.9 COMPUTER VISION MODEL DESIGN

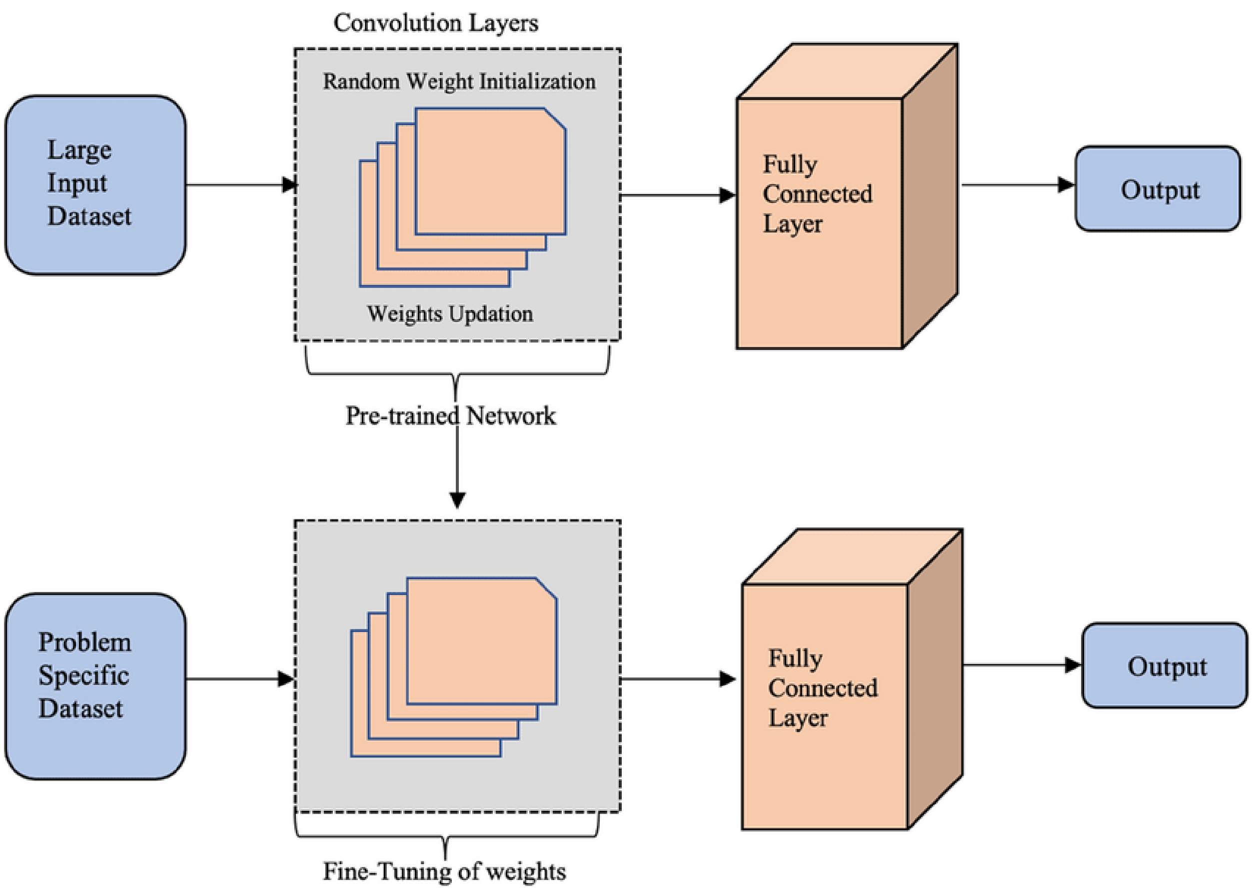


Figure 43. Architecture of transfer learning model (Sharma et. al.)

In the context of this study, transfer learning was predominantly utilized in the development of image classification services. The use of transfer learning significantly increased the efficacy of our image classification models and helped us overcome some of the challenges related to data and computation resources. The architecture of the Transfer Learning approach can be seen on Figure 43.

The pre-existing layers of a pre-trained model, trained on a large dataset such as ImageNet, were used as feature extractors. These features, instead of being learned from scratch, were transferable knowledge that provided a robust foundation for our model. Our approach involved fine-tuning these pre-existing layers to adapt to the specific image classification tasks of our research. The last few layers of the pre-trained model, responsible for classification, were customized according to our specific needs. This involved connecting it to a filly-connected neurons and adding Batch Normalization and Dropout layers to minimize overfitting. Corresponding output neurons were then added in the final layer to match the number of classes in our dataset and tailoring other hyperparameters to enhance the model's performance on our tasks.

The fine-tuning process was performed over several epochs, with the model progressively learning to classify images specific to our tasks with each iteration. With the help of transfer learning, we were able to build a model that could effectively classify images into categories specific to our research, despite having a limited amount of labeled data for each category.

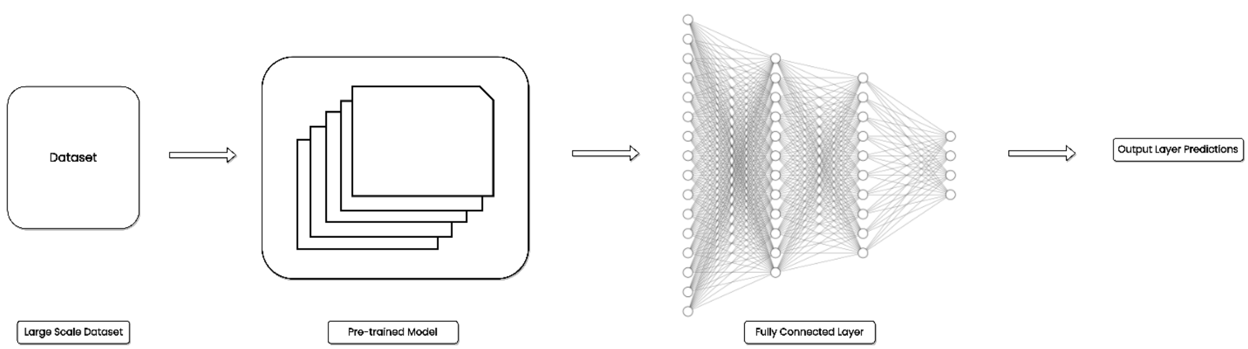


Figure 44. General Architecture for Image Classification Services

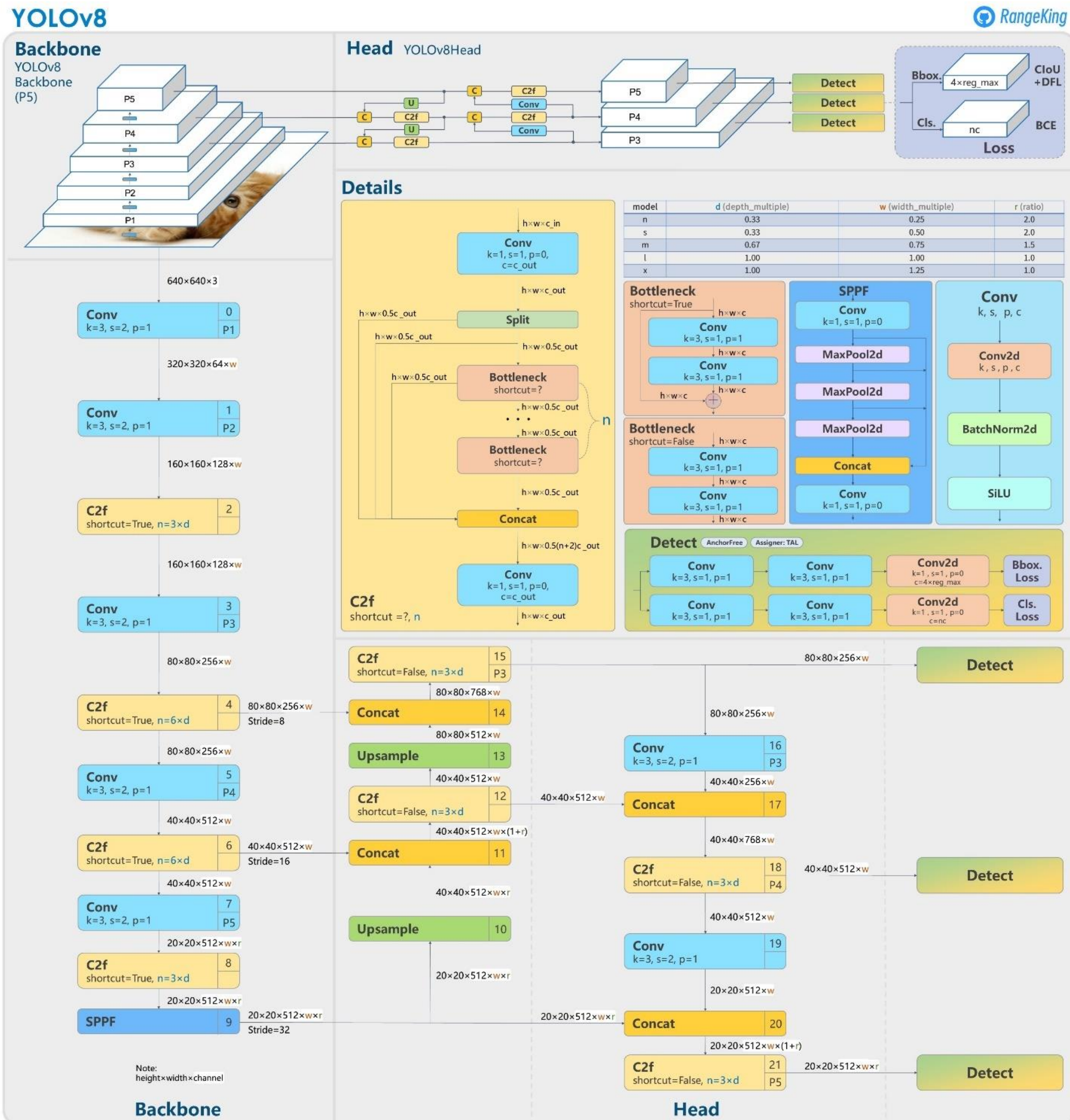


Figure 45. YOLOv8 Architecture

In this study, the YOLO (You Only Look Once) architecture was utilized for developing computer vision models. Specifically, the YOLOv8 version was implemented, which is renowned for its exceptional balance between speed and accuracy.

YOLO's approach to object detection is unique. Unlike the traditional two-step pipeline, where one stage is responsible for generating potential bounding boxes and another stage classifies those boxes, YOLO unifies this process into a single step. This methodology not only simplifies the computational flow but also significantly accelerates the detection process.

The YOLO architecture is composed of a single convolutional neural network (CNN) that takes an entire image as input and outputs bounding box coordinates and class probabilities directly in a single forward pass. This architecture can process images at remarkably high speeds, making it suitable for real-time object detection.

In the YOLOv8 model, several advancements are incorporated to improve detection accuracy, including:

1. Multi-Scale Predictions: YOLOv8 uses a multi-scale prediction strategy to make predictions at three different scales. This approach helps in detecting objects of various sizes present in the image.
2. Residual Connections: YOLOv8 employs residual connections (or shortcut connections) to prevent the vanishing gradient problem, thus enabling the training of deeper networks.
3. Anchor Boxes: The model uses anchor boxes to predict the aspect ratios of the objects, which assists in handling objects of different shapes and sizes.
4. DarkNet Architecture: YOLOv8 uses the DarkNet architecture, a custom-built CNN, as its backbone. It allows the network to run at high speed while retaining accuracy.

For our specific implementation, we utilized transfer learning, starting with a pre-trained YOLOv8 model. The pre-existing weights of this model, initially trained on a vast dataset, provided a robust starting point. The model was then fine-tuned on our specific datasets related to fish species, marine debris, and fishing vessels. The model's training involved adjustment of various hyperparameters including the number of epochs, image size, batch size, and the number of workers for data loading. The overall architecture of the YOLOv8 model can be found on Figure 45.

3.9.1 Custom Fish Vision

Table 11. *Custom Fish Vision Metadata*

Property	**Details**
Number of Images	30000
Number of Classes	4
Classes	Tilapia, Perch, Tarpon, Catfish
Count of Tilapia Images	14953
Count of Perch Images	15007
Count of Tarpon Images	15116
Count of Catfish Images	14859

Table 11 presents the detailed metadata of the custom Fish Vision dataset utilized in training the YOLOv8 model for this study. The dataset is composed of 30,000 images, evenly distributed across four distinct classes - Tilapia, Perch, Tarpon, and Catfish.

The YOLOv8 model, a state-of-the-art object detection algorithm, was utilized in the development of the custom fish vision service in. This model was chosen due to its high accuracy and efficiency in detecting and classifying objects in images. To tailor the model to Aquaria's specific requirements, it was trained on the dataset with an epoch count of 20, an image size of 512x512 pixels, a seed value of 42 for reproducibility, a batch size of 8, and a worker count of 4 for parallel data loading. By training the model on this extensive and balanced dataset, the study ensured the robustness and precision of the computer vision services provided by Aquaria.

Table 12. *Training Parameters of YOLOv8 Model for Custom Fish Vision Dataset*

Property	**Value**
Pretrained Model	yolov8x.pt
Data Path	/content/dataset.yaml
Number of Epochs	20
Image Size	512x512

Seed Value	42
Batch Size	8
Number of Workers	4

Table 12 outlines the configuration parameters used for training the model. It includes details about the pretrained model used, the dataset path, the number of epochs for which the model was trained, the size of the images used, the seed value for reproducibility, the batch size used, and the number of worker threads.

3.9.2 Deep Fish

Table 13. *Training Parameters of YOLOv8 Model for Deep Fish Dataset*

Parameter	Value
Pretrained model	yolov8x.pt
Data	/content/data.yaml
Epochs	40
Image Size	480x270
Seed	42
Batch Size	32
Workers	2

Table 13 represents the training parameters utilized for the Deep Fish Vision model. A pretrained model, 'yolov8x.pt', was leveraged as a starting point for training, capitalizing on the established knowledge encoded in this pre-existing model to accelerate and refine the training process. The dataset used for training was specified in '/content/data.yaml'.

The training process spanned 40 epochs, with each epoch representing a complete pass through the entire training dataset. An image size of 480x270 was used for the input images. The 'seed' parameter, set to 42, was used to ensure the reproducibility of the training results by providing a deterministic starting point for the random number generators used in the training process.

The batch size was set to 32, meaning that 32 samples from the training dataset were used for each update to the model's weights. The 'workers' parameter was set to 2, indicating the number of subprocesses used for data loading. This parameter is crucial in maintaining the efficiency of the training process by ensuring that the next batch of data is ready for use while the current batch is being processed.

3.9.3 Marine Debris

Table 14 depicts the training parameters used for the Marine Debris Vision model. As with the previous models, the 'yolov8x.pt' pretrained model was utilized as the initial foundation for training. This model was then fine-tuned on the dataset specified in '/content/Underwater_garbage/data.yaml', focusing on the task of marine debris detection.

The training process was extended over 150 epochs, indicating that the entire training dataset was processed 150 times. The image size was dynamically set to match the height and width of the input images. A 'seed' value of 42 was utilized to guarantee the reproducibility of the training outcomes, by setting a deterministic starting point for the random number generators used in the training process.

The batch size was established as 32, implying that 32 samples from the training dataset were employed for each update to the model's weights. The 'workers' parameter was set to 2, representing the number of subprocesses used for data loading. This parameter is crucial for the efficiency of the training process, ensuring that the subsequent batch of data is prepared for use while the current batch is being processed.

Table 14. *Training Parameters of YOLOv8 Model for Marine Debris Dataset*

Property	**Value**
Pretrained Model	yolov8x.pt
Dataset	/content/Underwater_garbage/data.yaml
Epochs	150
Image Size	416x416
Seed	42
Batch Size	32
Workers	2

3.9.4 Marine Garbage

Table 15. *Training Parameters of YOLOv8 Model for Marine Garbage Dataset*

Property	Value
Pretrained Model	yolov8x.pt
Dataset	/content/dataset.yaml
Epochs	50
Image Size	480x270
Seed	42
Batch Size	16
Workers	2

Table 15 outlines the parameters that were used to train the Marine Garbage Vision Model. This model employed the same 'yolov8x.pt' pretrained model as in the previous cases. The dataset path for this model was '/content/dataset.yaml'. This model was trained for 50 epochs, which denotes the number of complete passes through the training dataset. The image size utilized was (480, 270). The seed for random number generation was set at 42 to ensure consistent results. A batch size of 16 was used, which determines the number of training examples utilized in one iteration. The model was configured to use 2 workers for loading the data. The carefully selected parameters ensured optimal learning and performance for the model in identifying and classifying marine garbage.

3.9.5 Fishing Vessels

Table 16. *Training Parameters of YOLOv8 Model for Fishing Vessels Dataset*

Property	Value
Pretrained Model	yolov8x.pt
Dataset	/kaggle/input/ships-in-aerial-images/ships-aerial-images/data.yaml
Epochs	20
Image Size	768x768
Seed	42

Batch Size	8
Workers	4

Table 16 details the parameters used for training the Fishing Vessels Vision Model. Similar to previous models, it used the 'yolov8x.pt' pretrained model as its starting point. The dataset was provided from the Kaggle's "Ships in Aerial Images" competition. The model was trained for 20 epochs. The image height was used as the image size input parameter, signifying that the images were resized to the specific height while maintaining their original aspect ratio. The seed value was set to 42 to ensure the reproducibility of the model's results. The model was trained with a batch size of 8 and was configured to use 4 workers for loading the data. These parameters were meticulously chosen to ensure optimal learning and performance for the model in detecting and classifying fishing vessels.

3.9.6 Marine Animals

Table 17. *Data Generator Parameters for Marine Animals Dataset*

Parameter	Train Data	Validation Data	Test Data
Dataframe	train_df	val_df	test_df
x_col	'Filepath'	'Filepath'	'Filepath'
y_col	'Label'	'Label'	'Label'
Target Size	(224, 224)	(224, 224)	(224, 224)
Color Mode	'rgb'	'rgb'	'rgb'
Class Mode	'categorical'	'categorical'	'categorical'
Batch Size	32	32	32
Shuffle	True	True	False
Seed	42	42	42
Subset	'training'	-	-

Table 17 provides a comprehensive overview of the data generation parameters used for training, validation, and testing in the image classification task for marine animals. Each dataset is associated with a DataFrame ('train_df' for training data, 'val_df' for validation data, and 'test_df' for test data) that includes the file paths of the images ('Filepath') and the corresponding labels ('Label'). The target size for all images was set to 224x224 pixels and the color mode was set to 'rgb', indicating that the images were processed in their original color. All datasets used a 'categorical' class mode, reflecting the multiclass nature of the classification task.

The batch size, which is the number of samples processed before the model is updated, was consistently set at 32 for all datasets. The 'shuffle' parameter was set to True for the training and validation datasets to ensure randomness in the selection of batches, while it was set to False for the test dataset to maintain the order of the data for accurate evaluation. The 'seed' parameter was used for the random number generator in shuffling the training and validation datasets, ensuring reproducibility. Finally, the 'subset' parameter was specified only for the training data. The value 'training' denotes that this data was a part of the dataset used specifically for training the model.

Table 18. *Data Augmentation Step for Training with Marine Animals Dataset*

Sequence Order	**Augmentation Step**	**Parameters / Description**
1	Resizing	Resize to 224x224 pixels
2	Rescaling	Scale values between 0 and 1
3	RandomFlip	Horizontal flip
4	RandomRotation	Rotate by 10%
5	RandomZoom	Zoom by 10%
6	RandomContrast	Adjust contrast by 10%

The above table, Table 18, outlines the sequential steps involved in the data augmentation process for the marine animals classification dataset. The process begins with resizing the images to 224x224 pixels, which is the target size for the model. Next, the images are rescaled to have pixel values in the range of 0 to 1, a necessary preprocessing step for neural networks.

The following steps introduce random variations to the image data to enhance the model's ability to generalize from the training data. These variations include randomly flipping the images horizontally, randomly rotating the images by up to 10% in either direction, randomly zooming into the images by up to 10%, and adjusting the contrast of the images randomly by up to 10%. These transformations augment the diversity of the dataset, ensuring the model is exposed to a wider variety of data during training, improving its robustness and ability to handle real-world data.

Table 19. *Parameters of Pretrained MobileNetV3Large Model for the Marine Animals Dataset*

Parameter	**Value**	**Description**
input_shape	(224, 224, 3)	The shape of the images that the model will take as input.
include_top	False	Whether to include the fully-connected layer at the top of the network, which is responsible for classification.
weights	'imagenet'	The weights to initialize the model with. 'imagenet' specifies that the model should be initialized with weights trained on the ImageNet dataset.
pooling	'avg'	The pooling operation applied after the final convolutional layer. 'avg' specifies that average pooling should be used.
trainable	False	Specifies whether the weights of the pre-trained model can be updated during training. If False, the model acts as a fixed feature extractor, and only the weights of the added layers can be updated.

Table 19 outlines the parameters utilized when loading the pretrained MobileNetV3Large model for the image classification task in this study. The model was initialized with weights trained on the ImageNet dataset, a broad and diverse image dataset frequently used for image classification tasks. The model was configured to accept images of size 224x224x3, which corresponds to the image's height, width, and RGB color channels. The fully-connected layer at the top of the network, typically responsible for classification, was excluded, allowing for the custom configuration of the network's output layers. Average pooling was used as the pooling operation following the final convolutional layer, reducing the spatial dimensions of the output from these layers. The pretrained model was set to non-trainable, meaning the weights of the MobileNetV3Large model remained fixed during the training process, with only the weights of the newly added layers for this specific task updated. This technique is a key aspect of transfer learning, where a model's general features learned from a larger dataset are utilized for a specific task, often with less available data.

Table 20. *Customized Layers Added on Top of the Pre-Trained Model for the Marine Animals Dataset*

Parameter	Value	Description
Inputs	pretrained_model.input	The input tensor for the new model is the same as the input tensor of the pre-trained model.
Layer 1	Dense(256, relu)	A dense (fully connected) layer with 256 units and a rectified linear unit (ReLU) activation function.
Layer 2	Dropout(0.2)	A dropout layer with a rate of 0.2, which randomly sets a fraction (20%) of input units to 0 at each update during training.
Layer 3	Dense(256, relu)	Another dense layer with 256 units and a ReLU activation function.
Layer 4	Dropout(0.2)	Another dropout layer with a rate of 0.2.
Outputs	Dense(23, softmax)	The output layer is a dense layer with 4 units (corresponding to the 4 classes), with a softmax activation function for multiclass classification.

Table 20 presents the configuration of the fully connected layers added on top of the pretrained model. Two dense layers, each with 256 neurons and 'relu' as the activation function, are followed by a dropout layer with a rate of 0.2, to prevent overfitting. The final layer is a dense layer with 23 neurons corresponding to the 23 classes and uses 'softmax' as the activation function to provide a probability distribution over the classes.

Table 21. *Model Compilation Parameters for the Marine Animals Dataset*

Parameter	Value	Description
Optimizer	Adam	The optimization algorithm used to minimize the loss function.
Learning Rate	0.00001	The learning rate for the Adam optimizer.
Loss Function	Categorical Crossentropy	The loss function that measures the model's performance.
Metrics	Accuracy	The metric used to evaluate the model's performance.

Table 21 details the parameters used during the model's compilation stage. The Adam optimization algorithm with a learning rate of 0.00001 was used to minimize the categorical cross entropy loss function. The model's performance was evaluated based on accuracy.

Table 22. *Model Training Parameters for the Marine Animals Dataset*

Parameter	**Value**	**Description**
Training Data	train_images	The training data used to train the model.
Steps per Epoch	len(train_images)	The total number of steps (batches of samples) to yield from the generator before declaring one epoch finished and starting the next epoch.
Validation Data	val_images	The validation data used to evaluate the model at the end of each epoch.
Validation Steps	len(val_images)	Total number of steps (batches of samples) to yield from the validation_data generator before stopping at the end of every epoch.
Number of Epochs	100	The number of times the learning algorithm will work through the entire training dataset.
Callbacks	Early Stopping, Tensorboard, Checkpoint	A set of functions to be applied at given stages of the training procedure.

Table 22 summarizes the parameters used during the model training process. The model was trained on the 'train_images' dataset for 100 epochs, with the number of steps per epoch equal to the total number of training images. The 'val_images' dataset was used for validation at the end of each epoch, with the number of validation steps equal to the total number of validation images. Callbacks for early stopping, tensorboard, and model checkpointing were implemented during the training process to prevent overfitting, monitor training, and save the best model respectively.

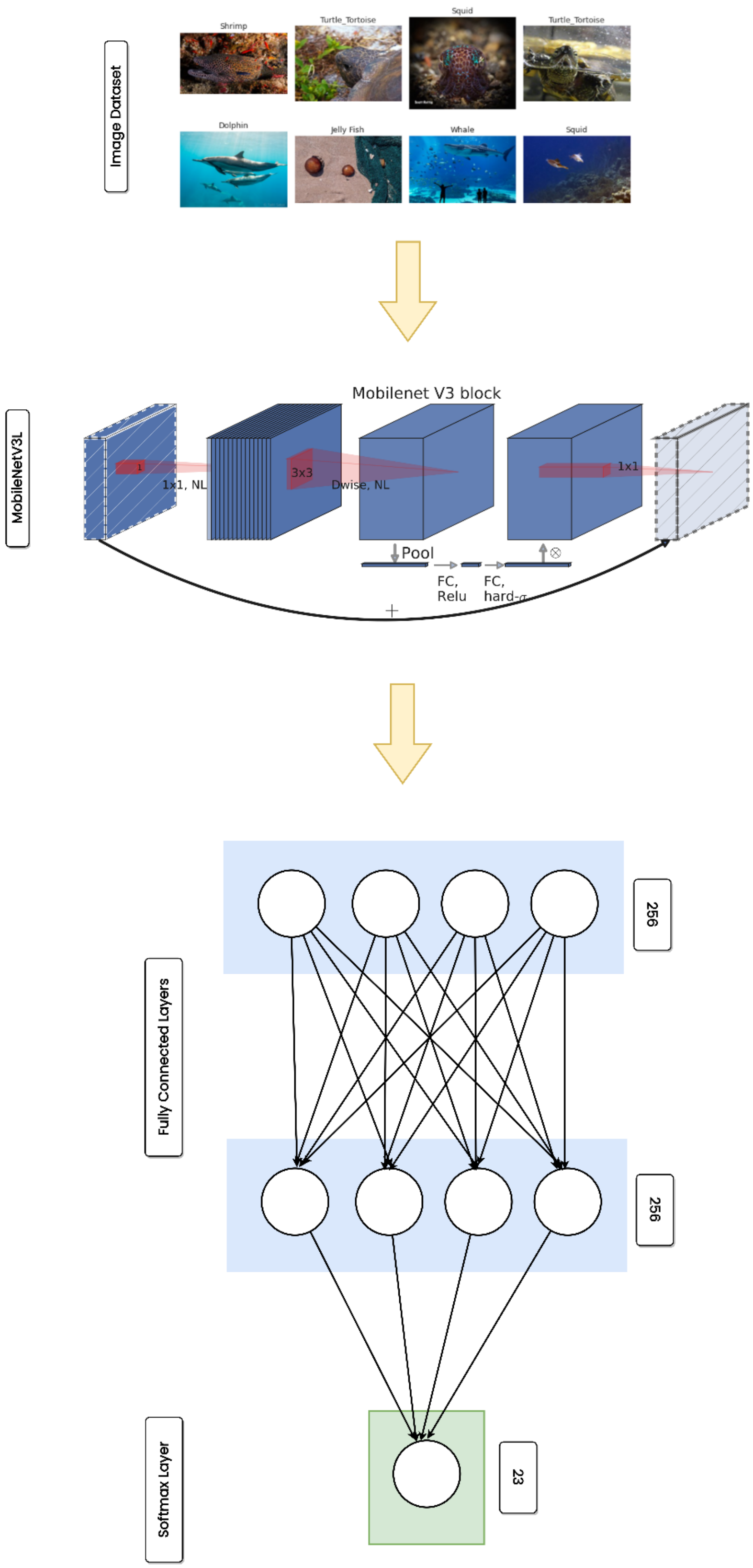


Figure 46. Marine Animals CNN Architecture

3.9.7 Market Marine Animals

Table 23 highlights the setup for the data generators for each data category (training, validation, and testing) in the Market Marine Animals dataset. Each data category was derived from a separate dataframe and used the 'Filepath' as the x_col and 'Label' as the y_col. The target size for the images was set to 224x224 pixels with an RGB color mode. The class mode was set as 'categorical', catering to the multi-class labels. For each category, a batch size of 32 was chosen. The Shuffle parameter was set to True for training and validation data, while False for testing data, ensuring consistent results. The Seed was uniformly set to 42 across all categories for reproducibility. The 'training' subset was uniquely applied to the training data category.

Table 23. *Data Generator Parameters for Market Marine Animals Dataset*

Parameter	**Train Data**	**Validation Data**	**Test Data**
Dataframe	train_df	val_df	test_df
x_col	'Filepath'	'Filepath'	'Filepath'
y_col	'Label'	'Label'	'Label'
Target Size	(224, 224)	(224, 224)	(224, 224)
Color Mode	'rgb'	'rgb'	'rgb'
Class Mode	'categorical'	'categorical'	'categorical'
Batch Size	32	32	32
Shuffle	True	True	False
Seed	42	42	42
Subset	'training'	-	-

Table 24. *Data Augmentation Steps for the Market Marine Animals Dataset*

Sequence Order	**Augmentation Step**	**Parameters / Description**
1	Resizing	Resize to 224x224 pixels
2	Rescaling	Scale values between 0 and 1

Table 24 showcases the sequential order of the data augmentation steps implemented on the Market Marine Animals dataset. The images were first resized to the dimensions of 224x224 pixels, in line with the input requirements of the pre-trained model. Following the resizing, rescaling was carried out to transform the pixel values to a range between 0 and 1, thereby standardizing the input data to the model.

Table 25. *Parameters of the Pretrained MobileNetV3Large Model for the Market Marine Animals Dataset*

Parameter	Value	Description
input_shape	(224, 224, 3)	The shape of the images that the model will take as input. The model expects input images to be 224x224 pixels in size, with 3 color channels (Red, Green, and Blue).
include_top	False	This parameter defines whether to include the fully-connected layer at the top of the network, which is responsible for classification. In this study, it is set to False, implying that this layer will not be included.
weights	'imagenet'	This parameter sets the weights the model will be initialized with. Setting this to 'imagenet' implies that the model will be initialized with weights trained on the ImageNet dataset, a large-scale dataset consisting of millions of labeled images across thousands of categories.
pooling	'avg'	The pooling operation to be applied after the final convolutional layer. Setting this to 'avg' indicates that average pooling should be used. Average pooling reduces the spatial dimensions of the output while preserving depth by taking the average of groups of inputs.

Table 25 outlines the key parameters for loading the pretrained MobileNetV3Large model, tailored for the Market Marine Animals dataset. By leveraging the weights of a model previously trained on the extensive ImageNet dataset, this study can capitalize on the model's ability to extract complex features from images. With the top fully connected layer excluded and training disabled, the pretrained model operates purely as a feature extractor, enabling the training process to focus on optimizing the weights of the newly added layers for the specific classification task at hand.

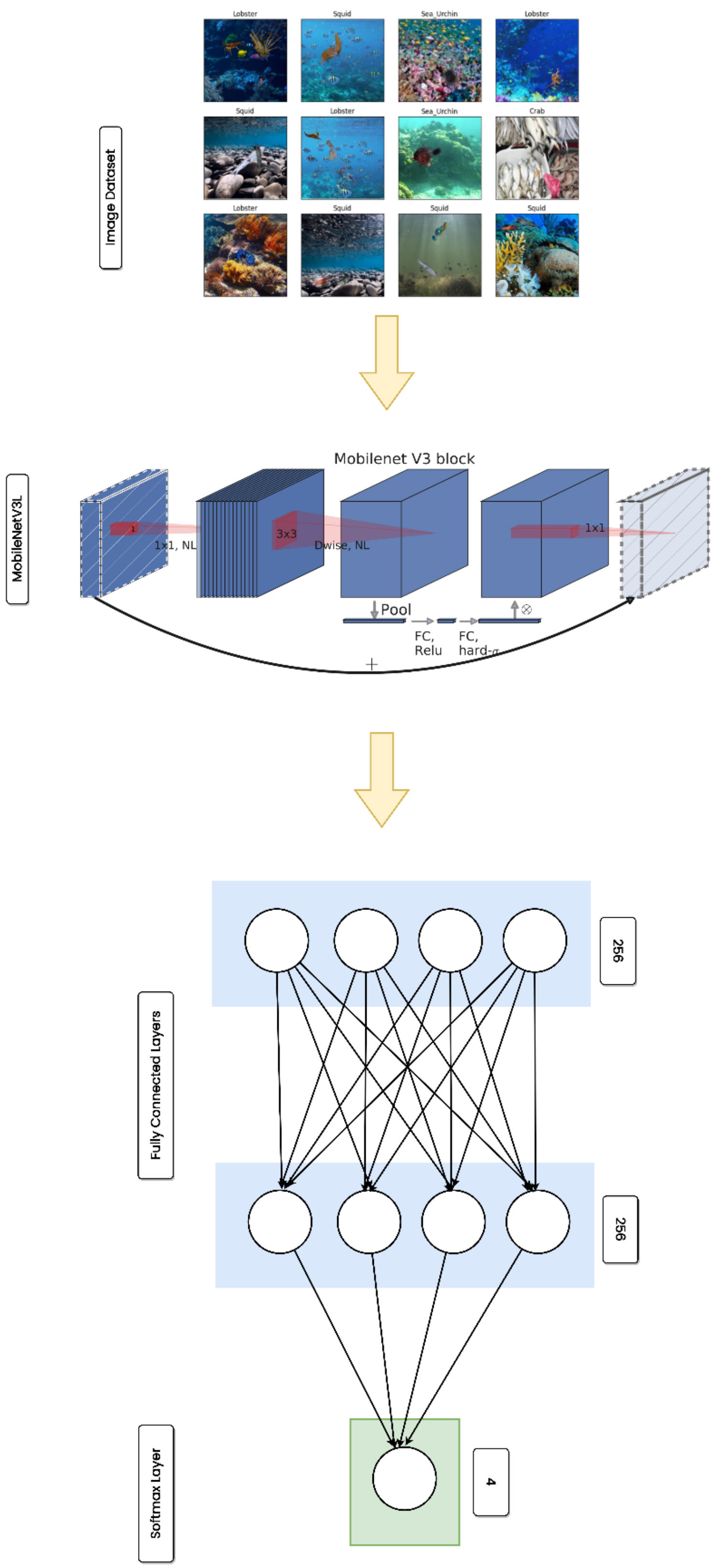


Figure 47. Market Marine Animals CNN Architecture

Table 26. *Customized Layers Added on Top of the Pre-Trained Model for the Market Marine Animals Dataset*

Parameter	Value	Description
Inputs	pretrained_model.input	The input tensor for the new model is the same as the input tensor of the pre-trained model.
Layer 1	Dense(256, activation='relu')	A dense (fully connected) layer with 256 units and a rectified linear unit (ReLU) activation function.
Layer 2	Dropout(0.2)	A dropout layer with a rate of 0.2, which randomly sets a fraction (20%) of input units to 0 at each update during training.
Layer 3	Dense(256, activation='relu')	Another dense layer with 256 units and a ReLU activation function.
Layer 4	Dropout(0.2)	Another dropout layer with a rate of 0.2.
Outputs	Dense(4, activation='softmax')	The output layer is a dense layer with 4 units (corresponding to the 4 classes), with a softmax activation function for multiclass classification.

Table 27. *Model Compilation Parameters for the Market Marine Animals Dataset*

Parameter	Value	Description
Optimizer	Adam(0.00001)	The optimizer used is the Adam optimizer, with a learning rate of 0.00001. Adam is an optimization algorithm that can handle sparse gradients on noisy problems, well suited for problems that are large in terms of data and/or parameters.
Loss Function	Categorical Cross entropy	The loss function used is categorical cross-entropy, which is appropriate for multiclass classification problems. It is a popular choice for classification tasks because it produces a score that summarizes the average difference between the actual and predicted probability distributions across all

		classes.
Metrics	accuracy	The metric used to evaluate model performance is classification accuracy, which calculates the proportion of correct predictions over all predictions.

Table 28. *Model Training Parameters for the Market Marine Animals Dataset*

Parameter	**Value**	**Description**
Training Data	train_images	The training data used is the augmented images from the train_images data generator.
Steps per Epoch	len(train_images)	The number of steps per epoch is set to the total number of samples in the training dataset divided by the batch size.
Validation Data	val_images	The validation data used is the augmented images from the val_images data generator.
Validation Steps	len(val_images)	The number of steps to validate at the end of each epoch is set to the total number of samples in the validation dataset divided by the batch size.
Epochs	100	The total number of iterations over the entire dataset is set to 100.
Callbacks	early_stopping, tensorboard, checkpoint	The model training uses early stopping, TensorBoard for generating logs, and model checkpoint callbacks. Early stopping is used to prevent overfitting by stopping the training process once the model's performance starts to degrade on the validation dataset.

Tables, 26, 27, and 28 give an in-depth understanding of how the new model is defined, compiled, and trained for the Market Marine Animals dataset. The model is based on the pre-trained MobileNetV3Large model with additional dense and dropout layers added on top to help generalize and prevent overfitting. The model is then trained using the Adam optimizer and categorical cross-entropy loss function, with accuracy being the metric for performance evaluation. Model training is performed over 100 epochs with early stopping for efficiency.

3.9.8 Trash Annotations in Context (TACO)

Table 29. *Training Parameters of YOLOv8 Model for TACO Dataset*

Property	**Value**
Pretrained Model	yolov8x.pt
Dataset	/content/data.yaml
Epochs	400
Image Size	416x416
Seed	42
Batch Size	128
Workers	2

The training process for the TACO YOLO model involved utilizing the "yolov8x.pt" pretrained model. The dataset used for training was sourced from "/content/data.yaml". The training was conducted over 400 epochs, with an image size of 416x416 pixels. A seed value of 42 was set to ensure reproducibility of results. The batch size was set to 128, and 2 workers were assigned for parallel processing during training. This configuration allowed for the model to learn and refine its detection capabilities using the TACO dataset.

3.10 EVALUATION AND INFERENCE METRICS

3.10.1 Dataset Metrics

Error Level Analysis (ELA)

Error Level Analysis (ELA) was utilized as a key methodology to identify potential disparities in compression levels across images. This technique served as an effective tool in the detection of image manipulation, based on the principle that modified regions of an image exhibit a distinct error level, having effectively been saved an additional time in the process of editing. The process involved the computation of differences between the original image and a re-compressed version of the image, with these differences subsequently enhanced for more effective visualization. The resulting ELA image provided a comprehensive illustration of error level variations across the image, thereby facilitating the detection of anomalies.

The success of the methodology was evaluated based on the degree to which manipulations were identifiable in the ELA image. In instances where the elements of the image blended seamlessly into the background, rendering them indistinguishable in the ELA image, the methodology was deemed successful. This served as a testament to the effectiveness of the image manipulation, particularly in the context of machine learning where the creation of realistic and diverse training samples is imperative.

The study also made effective use of random sampling in diversifying the evaluation of the ELA technique and ensuring the attainment of success criteria across different instances and conditions. In essence, the ELA was leveraged as an analytical tool in understanding the impact of image manipulations, thereby allowing for the evaluation of the quality of these alterations based on their integration into the original image content.

Table 30. *ELA Perceptual Metrics*

Property	**Expectations**
Edges	In the ELA result, similar edges should have comparable brightness. Both low-contrast and high-contrast edges should be visually comparable to one another. Low-contrast edges in an original photograph should be nearly as bright as high-contrast edges.
Texture	Under ELA, comparable textures should be colored similarly. A flat surface will probably yield a lower ELA result than areas with more surface texture.
Surfaces	All flat surfaces should have roughly the same coloring under ELA, regardless of the actual color of the surface.

3.10.2 Image Classification Metrics

Classification Reports

A classification report was utilized to appraise the performance of the classification model in the study. This tool furnished a summary of the model's performance on the dataset, encapsulating metrics like precision, recall, F1 score, and support for each class.

Precision, defined as the ratio of true positive predictions to the total positive predictions (Equation 2), measured the model's competency in accurately identifying positive instances.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

Recall, on the other hand, assessed the model's capacity to discover all positive instances within the dataset. It was computed as the proportion of true positive predictions to the sum of true positives and false negatives (Equation 3).

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

The F1 score, which is the harmonic mean of precision and recall (Equation 4), was employed as a unified metric to gauge the model's performance.

$$\text{F1} = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

Accuracy was computed as the ratio of correct predictions made by the model to the total number of predictions. This simple and intuitive metric is often employed as a baseline for comparison among different classification models. Nevertheless, it was important to note that accuracy could potentially be misleading in specific circumstances. For instance, in the event of class imbalance in the dataset (i.e., one class is significantly more prevalent than others), a model that consistently predicts the majority class would yield high accuracy, even if it demonstrated poor competence in identifying the minority class. In the relevant formula, TP stood for true positive, FP for false positive, TN for true negative, and FN for false negative.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

In addition to these metrics, the classification report also included the support for each class. Support represented the number of instances in the dataset that belonged to a particular class. To generate a classification report, a classification model was first fitted on the training data and subsequently utilized to make predictions on the test data. The predicted classes were then compared to the true classes to compute the various evaluation metrics.

Table 31. *Classification Reports Metric*

Metric	Formula	Description
Precision	TP / (TP + FP)	Measures the proportion of correctly identified positive instances out of all predicted positive instances.
Recall	TP / (TP + FN)	Measures the proportion of correctly identified positive instances out of all actual positive instances.
F1 Score	2 * (Precision * Recall) / (Precision + Recall)	Harmonic mean of Precision and Recall, often used as a comprehensive measure of a model's performance.
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Measures the proportion of correct predictions out of total predictions. Can be misleading for imbalanced datasets.
Support	N/A	The number of instances that belong to each class in the dataset.

Confusion Matrix

A confusion matrix was utilized to assess the performance of the classification algorithms employed in this research. It is a tabular representation designed to exhibit the quantity of correct and incorrect predictions rendered by a model on a dataset. Each matrix row embodied the instances of a predicted class, while each column stood for the instances of an actual class, forming an intersection of predicted and actual classifications.

The outcomes of the classification were divided into four categories: true positive, true negative, false positive, and false negative. A true positive occurred when the model correctly predicted a positive class, whereas a true negative happened when the model accurately predicted a negative class. False positives emerged when the model incorrectly ascribed a positive class, and false negatives arose when the model inaccurately projected a negative class.

The confusion matrix played a dual role in this research. First, it was used to derive a range of performance metrics such as precision, recall, and F1 score. Precision was defined as the ratio of correct positive predictions out of all positive predictions made by the model. Recall represented the proportion of correct positive predictions out of all actual positive instances. The F1 score, a harmonic mean of precision and recall, provided an overall measure of the model's performance in maintaining a balance between precision and recall.

Second, the confusion matrix shed light on the specific types of errors that the models produced. High instances of false negatives suggested that the model lacked sensitivity to certain data patterns. In contrast, many false positives indicated potential model overfitting to the training data, thereby hindering generalizability to the test data. Therefore, the confusion matrix served as an essential tool in refining the model and understanding its strengths and weaknesses.

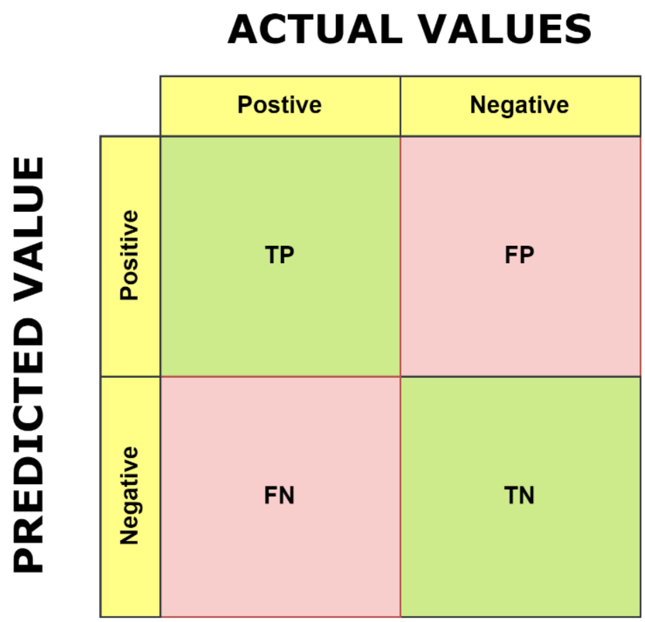


Figure 48. Confusion Matrix Structure

Gradient-weighted Class Activation Mapping

Grad-CAM, short for Gradient-weighted Class Activation Mapping, was employed in this research as a visualization technique to shed light on the areas within an image that the model focused on to make its predictions. The primary objective of this technique was to understand which regions of the image contributed the most to the model's final decision.

The output layer of the Image Classification model was connected to the final convolutional layer to extract the gradients flowing into the last convolutional layer. These gradients were then globally average-pooled to generate weights. The heatmap was obtained by the linear combination of activation maps (output of the last convolutional layer) and the computed weights. This heatmap was then resized to the size of the original image, providing a high-resolution, class-discriminative visualization.

This implementation of Grad-CAM was instrumental in achieving two key objectives. First, it was employed to debug and improve the model performance. By visualizing which parts of the image were focused on by the model, it was possible to comprehend if the model was basing its decisions on appropriate features. In instances where the model was not focusing on the correct features, further adjustments to the model or pre-processing of the data were made.

Second, Grad-CAM was used to improve the interpretability of the Image Classification model. By visualizing the regions of the image, the model found important for classification, a better understanding of the model's reasoning process was achieved. This transparency was particularly beneficial when presenting the results to stakeholders, as it allowed for a more intuitive understanding of how the model worked.

3.10.3 YOLO Metrics (Object Detection)

YOLO (You Only Look Once) is a popular object detection algorithm that computes bounding box coordinates and class predictions directly from a single pass of the image through the network. It uses a number of metrics to assess the model's performance, which are discussed below.

Box Loss

This represents the error in the predicted bounding box coordinates compared to the ground truth. In YOLO, a smaller box loss indicates better model performance because it means the predicted bounding boxes more accurately fit the actual objects in the image. It is calculated using a form of regression loss, such as Mean Squared Error (MSE).

Classification Loss (cls_loss)

This is the error in the predicted class labels compared to the ground truth. A lower classification loss implies that the model is more accurately classifying the objects within the detected bounding boxes. This is typically computed using Cross-Entropy Loss.

Center-ness Loss (dfl_loss)

This is a newer concept introduced in some variants of YOLO and other object detectors, which calculates the error in the predicted center-ness of objects within the bounding boxes. A lower center-ness loss implies that the object's center is more accurately localized within the bounding box.

Table 32. *YOLO Loss Metrics*

Metric	Description	Implication
Box Loss	Error in predicted bounding box coordinates	Lower is better; objects are more accurately located
Classification Loss (cls_loss)	Error in predicted class labels	Lower is better; objects are more accurately classified
Center-ness Loss (dfl_loss)	Error in predicted center-ness of objects	Lower is better; object centers are more accurately localized

Precision

This is the proportion of positive identifications that were actually correct. A model with high precision may not cover all positive samples, but what it predicts as positive are indeed positive.

Recall

This is the proportion of actual positive samples that were identified correctly. A model with high recall identifies most of the positive samples correctly but may also classify many negative samples as positive.

Mean Average Precision (mAP)

This metric calculates the average precision over different Intersection over Union (IoU) thresholds. It essentially summarises the precision-recall curve into a single value, making it a popular metric for comparing the performance of different object detectors.

Table 33. *YOLO Classification Metrics*

Metric	Meaning	Implication

Precision	Proportion of positive identifications that were correct	Higher is better; fewer false positives
Recall	Proportion of actual positives that were identified correctly	Higher is better; fewer false negatives
Mean Average Precision (mAP)	Average precision over different IoU thresholds	Higher is better; more accurate object detection

3.10.4 Service Oriented Architecture Quality Model (SOAQM) based on ISO 25010

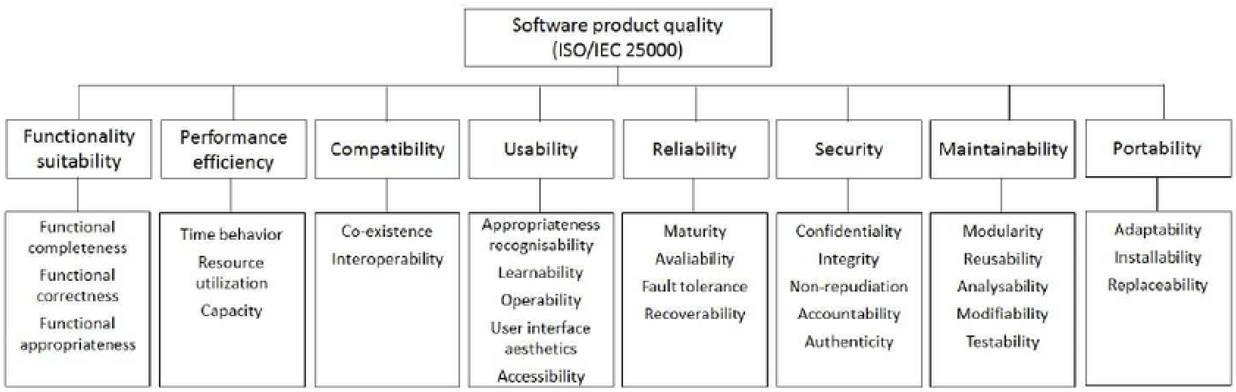


Figure 49. Quality model for external and internal quality by ISO 25010. Source: (M. S. França & S. Soares, 2015)

The evaluation of the service is designed around the Service Oriented Architecture Quality Model (SOAQM), which is based on ISO 25010. This model evaluates the service on multiple facets including functionality, efficiency, reliability, usability, maintainability, portability, and others. The application of the SOAQM ensures a thorough and comprehensive evaluation of the developed service, allowing the identification of potential areas of improvement (M. S. França & S. Soares, 2015).

Data collection for this research is conducted through a survey, aimed at gathering insights, opinions, and feedback about Aquaria from potential users and industry experts. The survey is an integral part of the research as it provides valuable user-oriented data, offering a unique perspective in evaluating the effectiveness, efficiency, and user-friendliness of the developed service. This user feedback serves as a critical component in the iterative process of development and refinement, guiding the adjustments to be made for enhancing the overall quality and effectiveness of Aquaria.

This methodology ensures a rigorous, user-centric, and iterative approach to the development and refinement of Aquaria, with the end goal of providing a valuable and efficient service to the local fishing and aquaculture industry in the Philippines.

Table 34. *ISO 25010 characteristics mapped to SOA quality characteristics. Source: (M. S. França & S. Soares, 2015)*

Characteristic	Sub-characteristics	SOA Perspective
Functional Suitability	*Functional completeness*	Services must cover all the specified tasks and user objectives which were designed
	Functional correctness	Services should provide the correct results with the needed degree of precision
	Functional appropriateness	Services are designed to facilitate accomplishment of specified tasks, more precisely, the execution of a business process.
Performance efficiency	*Time behavior*	Time spent by a service to process a request and return a response.
	Resource utilization	Services use resources such as servers to access information of other applications.
	Capacity	Service capacity can be defined as the ability to remain working even with large number of accesses at the same time.
Compatibility	*Co-existence*	Different composite services can share the use of same service operations.
	Interoperability	Various modules within a system, such as image processing and AI model inference, can exchange and effectively utilize the information that has been shared, in addition to the service's capacity to handle diverse

		types of image formats from user inputs.
Usability	*Appropriateness recognizability*	Users can recognize whether this service is appropriate for their needs. through service description that relate information such as service functionality and data types transmitted.
	Learnability	Degree to which a service can facilitate the understanding of its operation.
	Operability	A system exhibits operability through a user-friendly interface and comprehensive documentation that facilitate effective interaction and understanding of the service.
	User error protection	A system implements safeguards and mechanisms to prevent errors arising from inappropriate or unexpected user inputs. This involves incorporating input validation, data preprocessing steps, and providing clear user guidelines to mitigate the likelihood of user-induced errors and ensure a smoother user experience.
	User interface aesthetics	Not the focus of SOA.
	Accessibility	Not the focus of SOA.
Reliability	*Maturity*	Whenever a service consumer requests some information, it is expected that a response is returned.
	Availability	Services must be available when they are requested.
	Fault tolerance	Services can create strategies that may be performed when a failure happens on some hardware or software.

	Recoverability	Service ability to recover data when occurs some interruption or failure.
Security	*Confidentiality*	Information shared by a service provider can be accessed only to an authorized service client.
	Integrity	Services must be developed to prevent unauthorized access to, or modification of private data.
	Non-repudiation	Service provider constructs strategies to prove that an information have been delivered to a service consumer.
	Accountability	Services are autonomous.
	Authenticity	The identity of the external service provider should be authenticated.
Maintainability	*Modularity*	Service provider hosts a network accessible software module.
	Reusability	Services are reusable.
	Analyzability	Analyze change impact when services need to be modified
	Modifiability	Services are loosely coupled. This characteristic reduces the dependency between services, increasing modifiability.
	Testability	Services can be tested, for instance, through automated tools for functional testing
Portability	*Adaptability*	Although web services run remotely on a server, it can happen a change of platform.
	Installability	Although web services run remotely on a server, it can happen a change of platform.
	Replaceability	Although web services run remotely on a server, it can happen a change of platform.

3.11 DEPLOYMENT PLAN

3.11.1 Streamlit Deployment

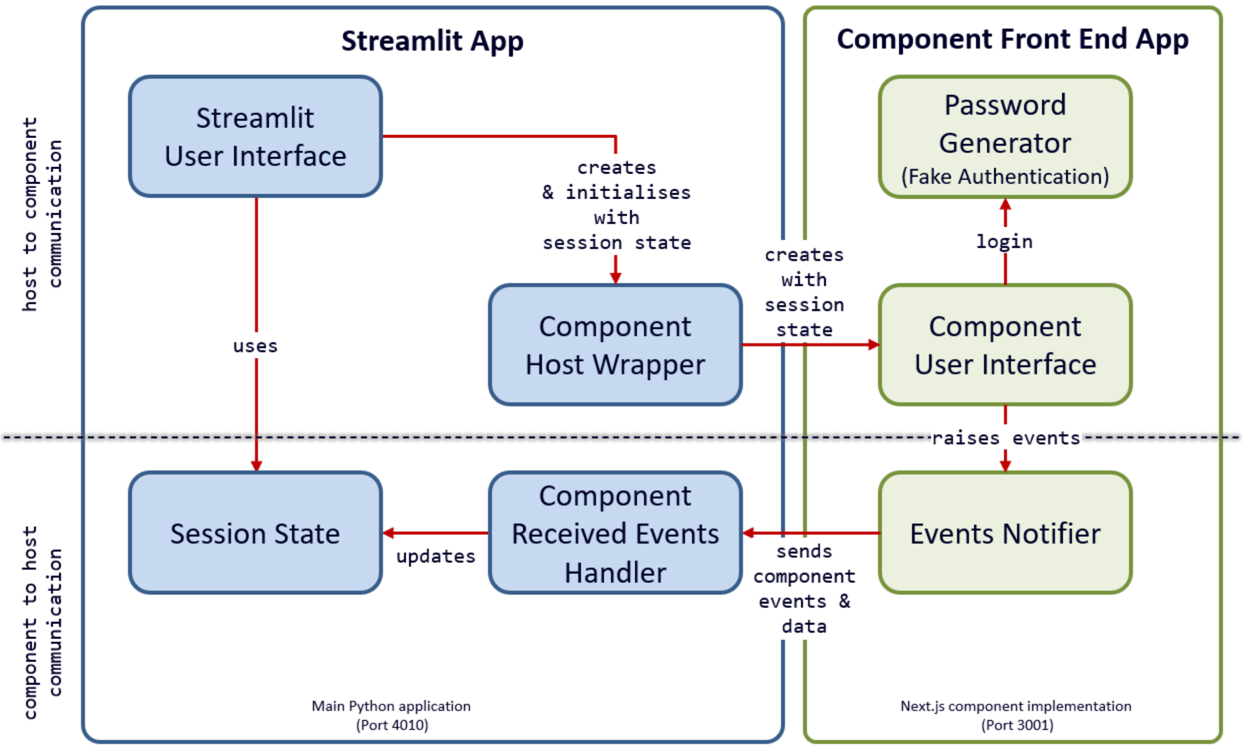


Figure 50. Streamlit Architecture (Streamlit, n.d.)

The deployment plan for Aquaria was a crucial part of the research process. Streamlit, a popular open-source Python library used to create custom web applications for machine learning and data science projects, played a key role in the deployment phase of Aquaria.

The deployment began by converting the functional image classification and object detection models into a user-friendly web application using Streamlit. This transformation enabled users without a deep technical background to easily interact with the models, making Aquaria more accessible.

Upon creating the web application, the next step in the deployment was hosting the application on Streamlit Cloud. Streamlit Cloud is a platform offered by Streamlit that allows developers to host and share their Streamlit applications on the internet. By hosting Aquaria on Streamlit Cloud, the application was made readily available to users worldwide, effectively increasing its reach and impact.

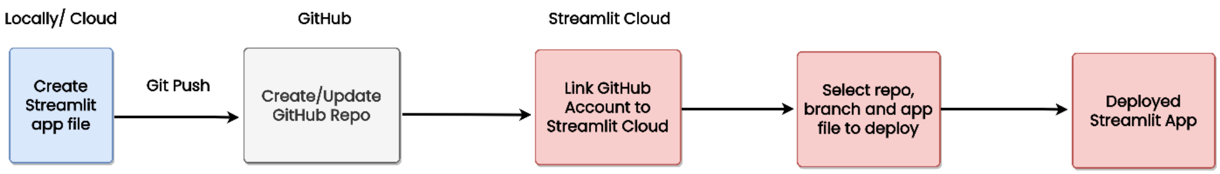


Figure 51. Streamlit Deployment Flow chart

Hosting on Streamlit Cloud also provided several other advantages. For instance, Streamlit Cloud ensured the seamless operation of the application by managing server setup, uptime, and maintenance. This allowed the researchers to focus on the functionality of the application rather than the technicalities of web hosting.

Additionally, Streamlit Cloud allowed for easy sharing of the application. A unique URL was generated for Aquaria, which could be shared with users, stakeholders, or other researchers. This feature facilitated easy dissemination of the research results and the practical application of the models developed.

3.11.2 Streamlit Web Interface

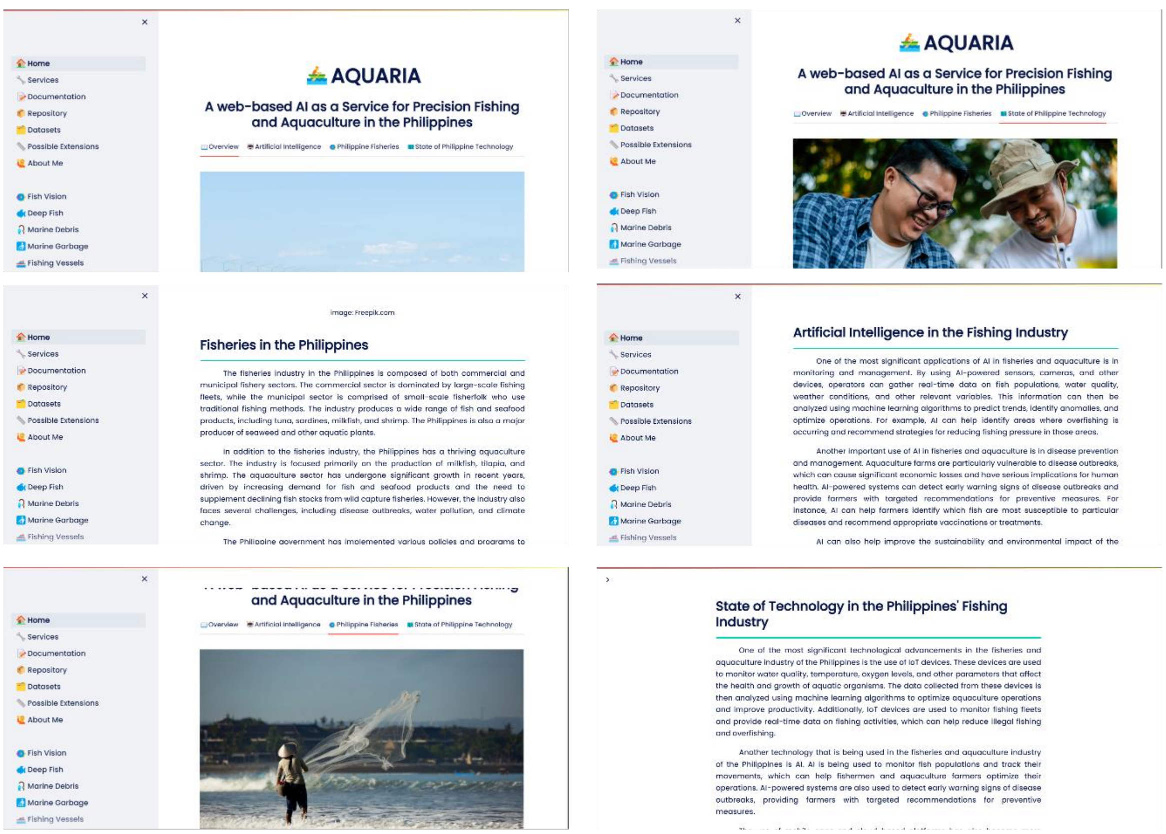


Figure 52. Streamlit Home Page Interface

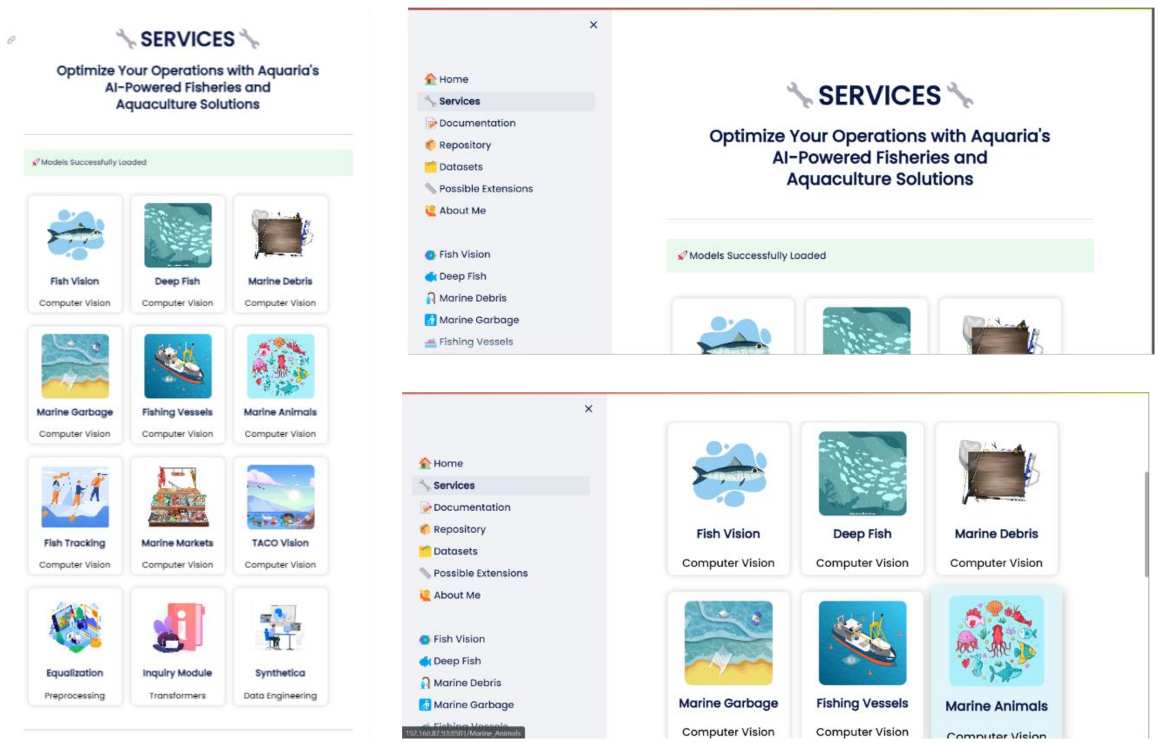


Figure 53. Streamlit Service Page Interface

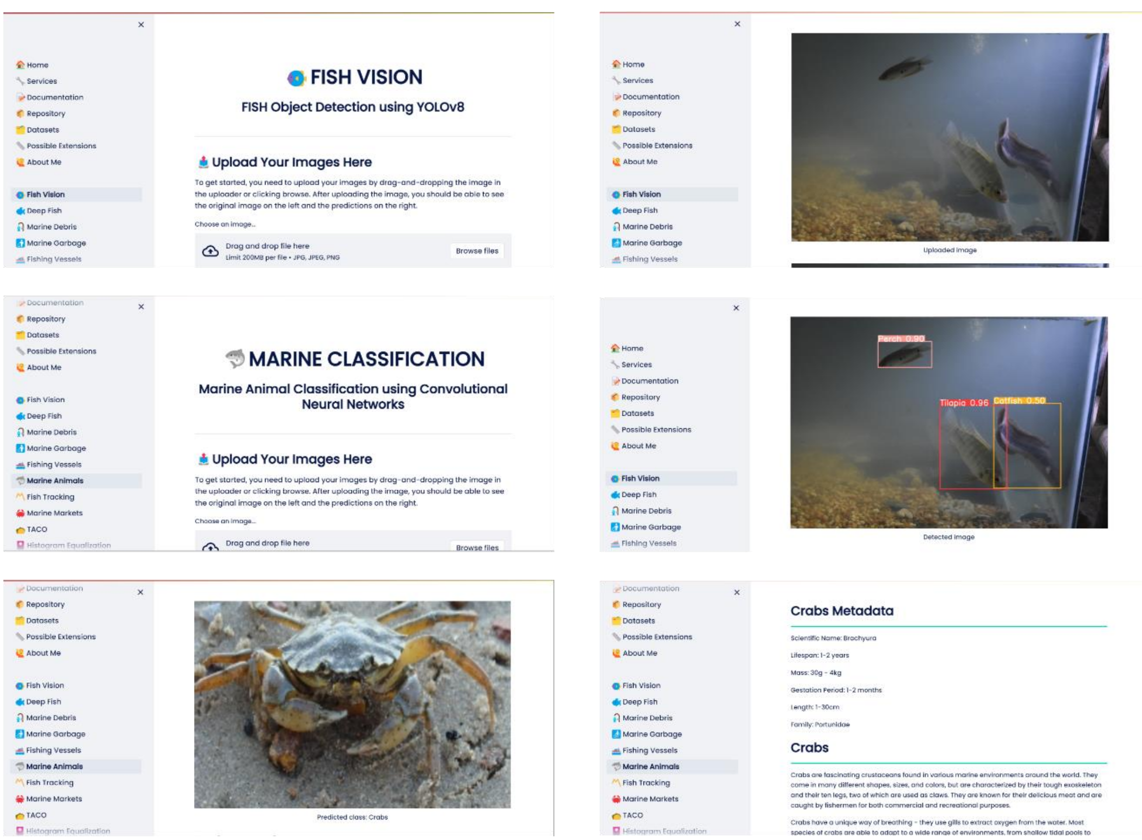


Figure 54. Sample Service Functionality

APPENDICES

A. REFERENCES

Abreo, N. A. S. (n.d.). *Marine plastics in the Philippines: A call for research*. 3.

Alimen, R., Alimen, Ma. C., Ortizo, C., Baynosa, R., & Corpes, V. (n.d.). Fishing Practices of the Community Folks and Sea Productivity in San Jose, Antique, Philippines. *IAMURE International Journal of Ecology and Conservation*, *6*(1). https://ejournals.ph/article.php?id=2977

Allken, V., Handegard, N. O., Rosen, S., Schreyeck, T., Mahiout, T., & Malde, K. (2019). Fish species identification using a convolutional neural network trained on synthetic data. *ICES Journal of Marine Science*, *76*(1), 342–349. https://doi.org/10.1093/icesjms/fsy147

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

An, D., Hao, J., Wei, Y., Wang, Y., & Yu, X. (2021). Application of computer vision in fish intelligent feeding system—A review. *Aquaculture Research*, *52*(2), 423–437. https://doi.org/10.1111/are.14907

Arcidiacono, C. (n.d.). *An empirical study on synthetic image generation techniques for object detectors*. 91.

Arlinghaus, R., & Cooke, S. J. (2005). Global Impact of Recreational Fisheries. *Science*, *307*(5715), 1561–1563. https://doi.org/10.1126/science.307.5715.1561

Asche, F., & Smith, M. D. (2018). Viewpoint: Induced Innovation in Fisheries and Aquaculture. *Food Policy*, *76*, 1–7. https://doi.org/10.1016/j.foodpol.2018.02.002

Belton, B., Little, D. C., Zhang, W., Edwards, P., Skladany, M., & Thilsted, S. H. (2020). Farming fish in the sea will not nourish the world. *Nature Communications*, *11*(1), 5804. https://doi.org/10.1038/s41467-020-19679-9

Bergland, H., Pedersen, P. A., & Wyller, J. (2019). Accumulated marine pollution and fishery dynamics. *Ecological Complexity*, *38*, 56–74. https://doi.org/10.1016/j.ecocom.2019.03.001

Bergmann, M. (Ed.). (2015). *Marine anthropogenic litter*. Springer.

Breckwoldt, A., Wang, W.-C., von Storch, H., & Ratter, B. M. W. (Eds.). (2019). *Fishing for Human Perceptions in Coastal and Island Marine Resource Use Systems, 2nd Edition*. Frontiers Media SA. https://doi.org/10.3389/978-2-88945-903-2

Brigato, L., & Iocchi, L. (2020). *A Close Look at Deep Learning with Small Data* (arXiv:2003.12843). arXiv. http://arxiv.org/abs/2003.12843

Bucol, A. A. (2016). *Diminishing sustainability of traditional fishing practices in Siquijor Island, central Philippines*. 11.

Burden, F., & Winkler, D. (2008). Bayesian Regularization of Neural Networks. In D. J. Livingstone (Ed.), *Artificial Neural Networks* (Vol. 458, pp. 23–42). Humana Press. https://doi.org/10.1007/978-1-60327-101-1_3

Campbell, S., Greenwood, M., Prior, S., Shearer, T., Walkem, K., Young, S., Bywaters, D., & Walker, K. (2020). Purposive sampling: Complex or simple? Research case examples. *Journal of Research in Nursing*, *25*(8), 652–661. https://doi.org/10.1177/1744987120927206

Cappo, M., Harvey, E., Malcolm, H., & Speare, P. (n.d.). *POTENTIAL OF VIDEO TECHNIQUES TO MONITOR DIVERSITY, ABUNDANCE AND SIZE OF FISH IN STUDIES OF MARINE PROTECTED AREAS*. 11.

Cappo, M., Harvey, E., & Shortis, M. (n.d.). *Counting and measuring fish with baited video techniques—An overview*. 15.

Chang, C.-C., Wang, Y.-P., & Cheng, S.-C. (2021). Fish Segmentation in Sonar Images by Mask R-CNN on Feature Maps of Conditional Random Fields. *Sensors*, *21*(22), 7625. https://doi.org/10.3390/s21227625

Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F. K., & Mahmood, F. (2021). Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, *5*(6), 493–497. https://doi.org/10.1038/s41551-021-00751-8

Cooke, S. J., Venturelli, P., Twardek, W. M., Lennox, R. J., Brownscombe, J. W., Skov, C., Hyder, K., Suski, C. D., Diggles, B. K., Arlinghaus, R., & Danylchuk, A. J. (2021). Technological innovations in the recreational fishing sector: Implications for fisheries management and policy. *Reviews in Fish Biology and Fisheries*, *31*(2), 253–288. https://doi.org/10.1007/s11160-021-09643-1

Costello, C., Ovando, D., Clavelle, T., Strauss, C. K., Hilborn, R., Melnychuk, M. C., Branch, T. A., Gaines, S. D., Szuwalski, C. S., Cabral, R. B., Rader, D. N., & Leland, A. (2016). Global fishery prospects under contrasting management regimes. *Proceedings of the National Academy of Sciences*, *113*(18), 5125–5129. https://doi.org/10.1073/pnas.1520420113

Daw, T. M., Cinner, J. E., McClanahan, T. R., Brown, K., Stead, S. M., Graham, N. A. J., & Maina, J. (2012). To Fish or Not to Fish: Factors at Multiple Scales Affecting Artisanal Fishers' Readiness to Exit a Declining Fishery. *PLoS ONE*, *7*(2), e31460. https://doi.org/10.1371/journal.pone.0031460

Dhar, P., & Guha, S. (2021). Fish Image Classification by XgBoost Based on Gist and GLCM Features. *International Journal of Information Technology and Computer Science*, *13*(4), 17–23. https://doi.org/10.5815/ijitcs.2021.04.02

dos Santos, P. S. M., Beltrão, A. C., de Souza, B. P., & Travassos, G. H. (2018). On the benefits and challenges of using kanban in software engineering: A structured synthesis study. *Journal of Software Engineering Research and Development*, *6*(1), 13. https://doi.org/10.1186/s40411-018-0057-1

Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 1310–1319. https://doi.org/10.1109/ICCV.2017.146

Food and Agriculture Organization of the United Nations [FAO]. (2020). *The State of World Fisheries and Aquaculture 2020*. FAO. https://doi.org/10.4060/ca9229en

French, G., Mackiewicz, M., Fisher, M., Holah, H., Kilburn, R., Campbell, N., & Needle, C. (2020). Deep neural networks for analysis of fisheries surveillance video and automated monitoring of fish discards. *ICES Journal of Marine Science*, *77*(4), 1340–1353. https://doi.org/10.1093/icesjms/fsz149

Frolov, V., Faizov, B., Shakhuro, V., Sanzharov, V., Konushin, A., Galaktionov, V., & Voloboy, A. (2022). Image Synthesis Pipeline for CNN-Based Sensing Systems. *Sensors*, *22*(6), 2080. https://doi.org/10.3390/s22062080

Georgakis, G., Mousavian, A., Berg, A. C., & Kosecka, J. (2017). *Synthesizing Training Data for Object Detection in Indoor Scenes* (arXiv:1702.07836). arXiv. http://arxiv.org/abs/1702.07836

Gilman, E., Musyl, M., Suuronen, P., Chaloupka, M., Gorgin, S., Wilson, J., & Kuczenski, B. (2021). Highest risk abandoned, lost and discarded fishing gear. *Scientific Reports*, *11*(1), 7195. https://doi.org/10.1038/s41598-021-86123-3

Green, S. J. (2003). *Philippine fisheries in crisis: A framework for management*. Coastal Resource Management Project.

Gunn, R., Hardesty, B. D., & Butler, J. (2010). Tackling 'ghost nets': Local solutions to a global issue in northern Australia: FEATURE. *Ecological Management & Restoration*, *11*(2), 88–98. https://doi.org/10.1111/j.1442-8903.2010.00525.x

Guo, Y., Li, Y., Wang, L., & Rosing, T. (2020). AdaFilter: Adaptive Filter Fine-Tuning for Deep Transfer Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4060–4066. https://doi.org/10.1609/aaai.v34i04.5824

Hampton-Smith, M., Bower, D. S., & Mika, S. (2021). A review of the current global status of blast fishing: Causes, implications and solutions. *Biological Conservation*, *262*, 109307. https://doi.org/10.1016/j.biocon.2021.109307

Hardesty, B. D., Roman, L., Duke, N. C., Mackenzie, J. R., & Wilcox, C. (2021). Abandoned, lost and discarded fishing gear 'ghost nets' are increasing through time in Northern Australia. *Marine Pollution Bulletin*, *173*, 112959. https://doi.org/10.1016/j.marpolbul.2021.112959

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

Honda, H. (2020, January 6). *Digging into Detectron 2—Part 1* [Blog]. Medium.Com. https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd

Hong, J., Fulton, M., & Sattar, J. (2020). *TrashCan: A Semantically-Segmented Dataset towards Visual Detection of Marine Debris* (arXiv:2007.08097). arXiv. http://arxiv.org/abs/2007.08097

Iqbal, U., Li, D., & Akhter, M. (2022). Intelligent Diagnosis of Fish Behavior Using Deep Learning Method. *Fishes*, *7*(4), 201. https://doi.org/10.3390/fishes7040201

Izhari, F., Zarlis, M., & Sutarman. (2020). Analysis of backpropagation neural neural network algorithm on student ability based cognitive aspects. *IOP Conference Series: Materials*

Science and Engineering, *725*(1), 012103. https://doi.org/10.1088/1757-899X/725/1/012103

Jambeck, J. R., Geyer, R., Wilcox, C., Siegler, T. R., Perryman, M., Andrady, A., Narayan, R., & Law, K. L. (2015). Plastic waste inputs from land into the ocean. *Science*, *347*(6223), 768–771. https://doi.org/10.1126/science.1260352

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, *199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Joaquin, H. D. (2021). *Traditional Fishing Practices: A Documentary*. *25*(4), 6.

Katikiro, R. E., & Mahenge, J. J. (2016). Fishers' Perceptions of the Recurrence of Dynamite-Fishing Practices on the Coast of Tanzania. *Frontiers in Marine Science*, *3*. https://doi.org/10.3389/fmars.2016.00233

Kawamura, G., & Bagarinao, T. (1980). *Fishing Methods and Gears in Panay Island, Philippines**. *29*, 42.

Kayri, M. (2016). Predictive Abilities of Bayesian Regularization and Levenberg–Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data. *Mathematical and Computational Applications*, *21*(2), 20. https://doi.org/10.3390/mca21020020

Keerthi, S., & Subhashini, R. (2022). Tracking of Fish Behaviour Using Computer Vision and Deep Learning. *Mathematical Statistician and Engineering Applications*, *Vol. 71 No. 3s2 (2022)*(2094–0343).

Kohtala, S., & Steinert, M. (2021). Leveraging synthetic data from CAD models for training object detection models – a VR industry application case. *Procedia CIRP*, *100*, 714–719. https://doi.org/10.1016/j.procir.2021.05.092

Kumar, A. S. (2021). *Image Segmentation and Object Recognition*. *1*, 12.

Lebreton, L. C. M., van der Zwet, J., Damsteeg, J.-W., Slat, B., Andrady, A., & Reisser, J. (2017). River plastic emissions to the world's oceans. *Nature Communications*, *8*(1), 15611. https://doi.org/10.1038/ncomms15611

Lee, H., Lee, J., Kim, H., & Mun, D. (2021). Dataset and method for deep learning-based reconstruction of 3D CAD models containing machining features for mechanical parts.

Journal of Computational Design and Engineering, *9*(1), 114–127. https://doi.org/10.1093/jcde/qwab072

Li, D., & Li, C. (2020). Intelligent aquaculture. *Journal of the World Aquaculture Society*, *51*(4), 808–814. https://doi.org/10.1111/jwas.12736

Li, J., Liu, C., Lu, X., & Wu, B. (2022). CME-YOLOv5: An Efficient Object Detection Network for Densely Spaced Fish and Small Targets. *Water*, *14*(15), 2412. https://doi.org/10.3390/w14152412

Liebelt, J., & Schmid, C. (2010). Multi-view object class detection with a 3D geometric model. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1688–1695. https://doi.org/10.1109/CVPR.2010.5539836

Liu, Z., Li, X., Fan, L., Lu, H., Liu, L., & Liu, Y. (2014). Measuring feeding activity of fish in RAS using computer vision. *Aquacultural Engineering*, *60*, 20–27. https://doi.org/10.1016/j.aquaeng.2014.03.005

M. S. França, J., & S. Soares, M. (2015). SOAQM: Quality Model for SOA Applications based on ISO 25010: *Proceedings of the 17th International Conference on Enterprise Information Systems*, 60–70. https://doi.org/10.5220/0005369100600070

Manettas, C., Nikolakis, N., & Alexopoulos, K. (2021). Synthetic datasets for Deep Learning in computer-vision assisted tasks in manufacturing. *Procedia CIRP*, *103*, 237–242. https://doi.org/10.1016/j.procir.2021.10.038

Mathiassen, J. R., Misimi, E., Østvik, S. O., & Aursand, I. G. (2012). Computer vision in the fish industry. In *Computer Vision Technology in the Food and Beverage Industries* (pp. 352–378). Elsevier. https://doi.org/10.1533/9780857095770.3.352

Nguyen, T.-A., Ly, H.-B., & Pham, B. T. (2020). Backpropagation Neural Network-Based Machine Learning Model for Prediction of Soil Friction Angle. *Mathematical Problems in Engineering*, *2020*, 1–11. https://doi.org/10.1155/2020/8845768

Niu, B., Li, G., Peng, F., Wu, J., Zhang, L., & Li, Z. (2018). Survey of Fish Behavior Analysis by Computer Vision. *Journal of Aquaculture Research & Development*, *09*(05). https://doi.org/10.4172/2155-9546.1000534

O'Donncha, F., & Grant, J. (2019). Precision Aquaculture. *IEEE Internet of Things Magazine*, *2*(4), 26–30. https://doi.org/10.1109/IOTM.0001.1900033

Palinkas, L. A., Horwitz, S. M., Green, C. A., Wisdom, J. P., Duan, N., & Hoagwood, K. (2015). Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Administration and Policy in Mental Health and Mental Health Services Research*, *42*(5), 533–544. https://doi.org/10.1007/s10488-013-0528-y

Pan, H. (2014). An Improved Back-Propagation Neural Network Algorithm. *Applied Mechanics and Materials*, *556–562*, 4586–4590. https://doi.org/10.4028/www.scientific.net/AMM.556-562.4586

Pandiyan, P., T, R., K, V. K., R, S., & T, T. (2020). Fish Species Classification using SVM Kernels. *Innovations in Information and Communication Technology Series*, 466–470. https://doi.org/10.46532/978-81-950008-1-4_102

Pauly, D., Christensen, V., Guénette, S., Pitcher, T. J., Sumaila, U. R., Walters, C. J., Watson, R., & Zeller, D. (2002). Towards sustainability in world fisheries. *Nature*, *418*(6898), 689–695. https://doi.org/10.1038/nature01017

Peng, X., Sun, B., Ali, K., & Saenko, K. (2015). Learning Deep Object Detectors from 3D Models. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1278–1286. https://doi.org/10.1109/ICCV.2015.151

Petrell, R. J., Shi, X., Ward, R. K., Naiberg, A., & Savage, C. R. (1997). Determining fish size and swimming speed in cages and tanks using simple video techniques. *Aquacultural Engineering*, *16*(1–2), 63–84. https://doi.org/10.1016/S0144-8609(96)01014-X

Petrov, A., & Popov, A. (2020). Overview of the application of computer vision technology in fish farming. *E3S Web of Conferences*, *175*, 02015. https://doi.org/10.1051/e3sconf/202017502015

Pham, V., Pham, C., & Dang, T. (2020). *Road Damage Detection and Classification with Detectron2 and Faster R-CNN* (arXiv:2010.15021). arXiv. http://arxiv.org/abs/2010.15021

Philippine Statistics Authority [PSA]. (2022). *Fisheries Situation Report for Major Species* (ISSN-2012-0400, p. 36) [Quarterly Report]. Philippine Statistics Authority [PSA]. https://psa.gov.ph/fisheries-situationer#:~:text=Commercial%20fisheries%20recorded%20a%20volume,of%20the%20total%20fisheries%20production.

Pinto, N., Barhomi, Y., Cox, D. D., & DiCarlo, J. J. (2011). Comparing state-of-the-art visual features on invariant object recognition tasks. *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, 463–470. https://doi.org/10.1109/WACV.2011.5711540

Ponnusamy, R., Sathyamoorthy, S., & Manikandan, K. (2017). A Review of Image Classification Approaches and Techniques. *International Journal of Recent Trends in Engineering and Research*, *3*(3), 1–5. https://doi.org/10.23883/IJRTER.2017.3033.XTS7Z

Qiu, C., Zhang, S., Wang, C., Yu, Z., Zheng, H., & Zheng, B. (2018). Improving Transfer Learning and Squeeze- and-Excitation Networks for Small-Scale Fine-Grained Fish Image Classification. *IEEE Access*, *6*, 78503–78512. https://doi.org/10.1109/ACCESS.2018.2885055

Ramdan, A., Heryana, A., Arisal, A., Kusumo, R. B. S., & Pardede, H. F. (2020). Transfer Learning and Fine-Tuning for Deep Learning-Based Tea Diseases Detection on Small Datasets. *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 206–211. https://doi.org/10.1109/ICRAMET51080.2020.9298575

Rana, S., Hasan, Md. N., Bari, A. A., Shimul, S. A., Ahmed, Sk. I., & Nahid, Sk. A. A. (2022). Problems and prospects of fish farming in the Chattogram Hill Tracts of Bangladesh: Community-based aquaculture might be a right choice. *Aquaculture and Fisheries*, S2468550X22000715. https://doi.org/10.1016/j.aaf.2022.04.002

Raza, K., & Hong, S. (2020). Fast and Accurate Fish Detection Design with Improved YOLO-v3 Model and Transfer Learning. *International Journal of Advanced Computer Science and Applications*, *11*(2). https://doi.org/10.14569/IJACSA.2020.0110202

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (arXiv:1506.02640). arXiv. http://arxiv.org/abs/1506.02640

Rekha, B. S., Srinivasan, G. N., Reddy, S. K., Kakwani, D., & Bhattad, N. (2020). Fish Detection and Classification Using Convolutional Neural Networks. In S. Smys, J. M. R. S. Tavares, V. E. Balas, & A. M. Iliyasu (Eds.), *Computational Vision and Bio-Inspired Computing* (Vol. 1108, pp. 1221–1231). Springer International Publishing. https://doi.org/10.1007/978-3-030-37218-7_128

Rodríguez, Á., Bermúdez, M., Rabuñal, J. R., & Puertas, J. (2015). Fish tracking in vertical slot fishways using computer vision techniques. *Journal of Hydroinformatics*, *17*(2), 275–292. https://doi.org/10.2166/hydro.2014.034

Rodriguez, A., Rico-Diaz, A. J., Rabuñal, J. R., Puertas, J., & Pena, L. (2015). Fish Monitoring and Sizing Using Computer Vision. In J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, Fco. J. Toledo-Moreo, & H. Adeli (Eds.), *Bioinspired Computation in Artificial Systems* (Vol. 9108, pp. 419–428). Springer International Publishing. https://doi.org/10.1007/978-3-319-18833-1_44

Romero, M., Interian, Y., Solberg, T., & Valdes, G. (n.d.). *Training Deep Learning models with small datasets*. 20.

SafetyNet Technologies. (2021, May 27). WITH PRECISION FISHING, HUMANS AND THE OCEAN CAN THRIVE TOGETHER. *WITH PRECISION FISHING, HUMANS AND THE OCEAN CAN THRIVE TOGETHER*. https://sntech.co.uk/blog/precision-fishing/

Saini, L. M. (n.d.). *Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton*. 7.

Saleh, A., Laradji, I. H., Konovalov, D. A., Bradley, M., Vazquez, D., & Sheaves, M. (2020). A Realistic Fish-Habitat Dataset to Evaluate Algorithms for Underwater Visual Analysis. *Scientific Reports*, *10*(1), 14671. https://doi.org/10.1038/s41598-020-71639-x

Sekhar, C., & Meghana, P. S. (2020). A Study on Backpropagation in Artificial Neural Networks. *Asia-Pacific Journal of Neural Networks and Its Applications*, *4*(1), 21–28. https://doi.org/10.21742/AJNNIA.2020.4.1.03

Shah, D. (2023, March 2). What is Synthetic Data in Machine Learning and How to Generate It [Blog]. *What Is Synthetic Data in Machine Learning and How to Generate It*. https://www.v7labs.com/blog/synthetic-data-guide

Sharma, R., Saqib, M., Lin, C. T., & Blumenstein, M. (2022). A Survey on Object Instance Segmentation. *SN Computer Science*, *3*(6), 499. https://doi.org/10.1007/s42979-022-01407-3

Siddagangu, S., Sujay, S., & Irfan, P. (2021). Fish Detection and Species Classification using Mask RCNN. *JETIR*, *8*(7), 5.

Sonatafy Technology. (n.d.). *PYTHON DEVELOPMENT*. EVERYTHING YOU WANTED TO KNOW ABOUT PYTHON DEVELOPMENT. Retrieved March 22, 2023, from https://sonatafy.com/python-development/

South East Asian Fisheries Development Center [SEAFDEC]. (n.d.). *Fisheries Country Profile: Philippines (2022)* [Country Profile]. South East Asian Fisheries Development Center [SEAFDEC]. Retrieved October 29, 2022, from http://www.seafdec.org/fisheries-country-profile-philippines-2022/

Stanford University. (n.d.). *CS231n Convolutional Neural Network for Visual Recognition*. CS231n Convolutional Neural Network for Visual Recognition. https://cs231n.github.io/convolutional-networks/

Stark, M., Goesele, M., & Schiele, B. (2010). Back to the Future: Learning Shape Models from 3D CAD Data. *Procedings of the British Machine Vision Conference 2010*, 106.1-106.11. https://doi.org/10.5244/C.24.106

Streamlit. (n.d.). Streamlit. Retrieved March 22, 2023, from https://streamlit.io/

Sun, M., Su, H., Savarese, S., & Fei-Fei, L. (n.d.). *A Multi-View Probabilistic Model for 3D Object Classes*. 8.

TahiLuddiN, A., & Terzi, E. (2021). An Overview of Fisheries and Aquaculture in the Philippines. *Journal of Anatolian Environmental and Animal Sciences*. https://doi.org/10.35229/jaes.944292

Taylor, S. M., Webley, J. A. C., & Mayer, D. G. (2011). Improving the precision of recreational fishing harvest estimates using two-part conditional general linear models. *Fisheries Research*, *110*(3), 408–414. https://doi.org/10.1016/j.fishres.2011.05.001

Tensorflow.org. (n.d.). *TensorFlow*. Retrieved March 22, 2023, from https://www.tensorflow.org/

Tian, Y., Li, X., Wang, K., & Wang, F.-Y. (2017). *Training and Testing Object Detectors with Virtual Images* (arXiv:1712.08470). arXiv. http://arxiv.org/abs/1712.08470

Tolentino-Zondervan, F., & Zondervan, N. A. (2022). Sustainable fishery management trends in Philippine fisheries. *Ocean & Coastal Management*, *223*, 106149. https://doi.org/10.1016/j.ocecoaman.2022.106149

Tonachella, N., Martini, A., Martinoli, M., Pulcini, D., Romano, A., & Capoccioni, F. (n.d.). OPEN An affordable and easy-to-use tool for automatic fish length and weight estimation in mariculture. *Scientific Reports*, 12.

van Emmerik, T., van Klaveren, J., Meijer, L. J. J., Krooshof, J. W., Palmos, D. A. A., & Tanchuling, M. A. (2020). Manila River Mouths Act as Temporary Sinks for Macroplastic Pollution. *Frontiers in Marine Science*, *7*, 545812. https://doi.org/10.3389/fmars.2020.545812

Republic Act No. 8550, (testimony of Cynthia Villar). Retrieved March 8, 2023, from https://lawphil.net/statutes/repacts/ra1998/ra_8550_1998.html

Wageeh, Y., Mohamed, H. E.-D., Fadl, A., Anas, O., ElMasry, N., Nabil, A., & Atia, A. (2021). YOLO fish detection with Euclidean tracking in fish farms. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 5–12. https://doi.org/10.1007/s12652-020-02847-6

Waldichuk, M. (1974). Coastal marine pollution and fish. *Ocean Management*, *2*(1), 1–60. https://doi.org/10.1016/0302-184X(74)90009-2

Wang, C., Li, Z., Wang, T., Xu, X., Zhang, X., & Li, D. (2021). Intelligent fish farm—The future of aquaculture. *Aquaculture International*, *29*(6), 2681–2711. https://doi.org/10.1007/s10499-021-00773-8

Wu, H., Zhou, Y., Luo, Q., & Basset, M. A. (2016). Training Feedforward Neural Networks Using Symbiotic Organisms Search Algorithm. *Computational Intelligence and Neuroscience*, *2016*, 1–14. https://doi.org/10.1155/2016/9063065

Wu, Y., Massa, F., Girshick, R., Kirillov, A., & Lo, W.-Y. (2019, October 10). Detectron2: A PyTorch-based modular object detection library [Blog]. *Detectron2: A PyTorch-Based Modular Object Detection Library*. https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

Yang, X., Zhang, S., Liu, J., Gao, Q., Dong, S., & Zhou, C. (2021). Deep learning for smart fish farming: Applications, opportunities and challenges. *Reviews in Aquaculture*, *13*(1), 66–90. https://doi.org/10.1111/raq.12464

Yao, H., Duan, Q., Li, D., & Wang, J. (2013). An improved K -means clustering algorithm for fish image segmentation. *Mathematical and Computer Modelling*, *58*(3–4), 790–798. https://doi.org/10.1016/j.mcm.2012.12.025

Yuan, B., Yue, F., Wang, X., & Xu, H. (2021). The Impact of Pollution on China Marine Fishery Culture: An Econometric Analysis of Heterogeneous Growth. *Frontiers in Marine Science*, *8*, 760539. https://doi.org/10.3389/fmars.2021.760539

Zhao, S., Zhang, S., Liu, J., Wang, H., Zhu, J., Li, D., & Zhao, R. (2021). Application of machine learning in intelligent fish aquaculture: A review. *Aquaculture*, *540*, 736724. https://doi.org/10.1016/j.aquaculture.2021.736724

Zhou, C., Lin, K., Xu, D., Chen, L., Guo, Q., Sun, C., & Yang, X. (2018). Near infrared computer vision and neuro-fuzzy model-based feeding decision system for fish in aquaculture. *Computers and Electronics in Agriculture*, *146*, 114–124. https://doi.org/10.1016/j.compag.2018.02.006

Zhou, R., Wu, D., Fang, L., Xu, A., & Lou, X. (2018). A Levenberg–Marquardt Backpropagation Neural Network for Predicting Forest Growing Stock Based on the Least-Squares Equation Fitting Parameters. *Forests*, *9*(12), 757. https://doi.org/10.3390/f9120757

Zhou, X. (2018). Understanding the Convolutional Neural Networks with Gradient Descent and Backpropagation. *Journal of Physics: Conference Series*, *1004*, 012028. https://doi.org/10.1088/1742-6596/1004/1/012028

Zion, B., Shklyar, A., & Karplus, I. (1999). Sorting fish by computer vision. *Computers and Electronics in Agriculture*, *23*(3), 175–187. https://doi.org/10.1016/S0168-1699(99)00030-7

B. QUESTIONNAIRE

Aquaria AI Web-Service SOAQM based on ISO 25010

Privacy Notice

At Lyceum of the Philippines University - Manila, we prioritize the protection of your privacy and strive to uphold it throughout the processing of your personal data.

For the purpose of conducting a research survey, we will collect specific information including your name, age, sex, and occupation. Additionally, it is optional to provide the number of years you have spent in your respective industry.

We are committed to ensuring the security and confidentiality of your personal data, taking all necessary precautions to prevent loss, misuse, and unauthorized processing activities. Your information will not be disclosed, shared, or transferred to any third party without your explicit consent.

By participating in this research survey and providing your personal information, you acknowledge and certify that you have read and understood the terms and conditions of this Privacy Notice. You confirm that the information you provide is accurate and true to the best of your knowledge. Please note that any decision made based on the information provided may be revised if it is found to be false or inaccurate.

In the event of any disputes or concerns related to the processing of your personal information, we are committed to resolving them amicably. However, if required, appropriate arbitration or court proceedings within the applicable jurisdiction will be pursued.

Your participation in this research survey is entirely voluntary, and you have the right to withdraw your consent or request the deletion of your personal data at any time. To exercise these rights or for any inquiries or concerns regarding your privacy, please contact us at privacy.manila@lpu.edu.ph.

Thank you for your willingness to contribute to our research. Your cooperation is greatly appreciated.

Lyceum of the Philippines University - Manila
College of Technology

1. Choice *

- ○ I Agree
- ○ I Disagree

Overview

Greetings,

My name is Lanz Vincent T. Vencer, and I am a 4th year BS Computer Science student specializing in Software Engineering at the Lyceum of the Philippines University - Manila. As part of my thesis, I am conducting a research study entitled "Aquaria: A Web-Based AI as a Service for Precision Fishing and Aquaculture in the Philippines Using Cut, Paste, and Learn Image Synthesis and Computer Vision Algorithms".

This survey aims to gather your valuable insights, opinions, and feedback about my proposed system. Your perspective is crucial in evaluating and enhancing the effectiveness, efficiency, and user- friendliness of Aquaria, thereby optimizing its impact on the local fishing and aquaculture industry.

The survey will base its model on the SOAQM: Quality Model for SOA Applications Based on ISO 25010, as introduced by Joyce Franca and Michel S. Soares from Universidade Federal de Uberlândia and Universidade Federal de Sergipe respectively.

Your participation is voluntary and all information provided will be kept confidential. Please feel free to stop the survey at any time.

I appreciate your time and input. Your responses will play an integral part in the success of this research project.

Thank you for your contribution. Should you have any questions or clarifications, feel free to contact me at lanz.vencer@lpunetwork.edu.ph.

Best regards,

Lanz Vincent T. Vencer

Researcher & 4th Year BSCS Undergraduate at LPU – Manila

Personal Information

2. Name

3. Age

4. Sex

5. Occupation/Field of Expertise *

- ☐ Aquaculture
- ☐ Scientist/Researcher
- ☐ Fisheries Biologist
- ☐ Fishery/Aquaculture
- ☐ Technician
- ☐ Fish Farm Manager
- ☐ Fisheries/Aquaculture
- ☐ Extension Officer
- ☐ Marine Biologist
- ☐ Data Scientist/Analyst
- ☐ Computer Vision Specialist
- ☐ Software Developer/Engineer
- ☐ Machine Learning Engineer
- ☐ Academe Professor
- ☐ Student
- ☐ Other

6. Industry Experience (Optional)

Functional Suitability

Sub-characteristic **Functional Completeness** means the degree to which the set of functions coversall the specified tasks and user objectives. Observing from the SOA perspective, services must cover all the specified tasks and user objectives for which they were designed.

Sub-characteristic **Functional Correctness** means the degree to which a product or system provides correct results with the needed degree of precision. This sub-characteristic can be applied in SOA by the fact that services should provide the correct response with the needed degree of precision.

Sub-characteristic **Functional Appropriateness** is the degree to which the functions facilitate the accomplishment of specified tasks and objectives. In the SOA context, services are designed to facili-tate the accomplishment of specified tasks, more precisely, the execution of a business process.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

7. Functional Completeness: The services cover all the specified tasks anduser objectives that were designed. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

8. **Functional Correctness**: The services exhibit functional correctness byproviding the correct results with the needed degree of precision. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

9. **Functional Appropriateness**: The services demonstrate functional appropriateness in facilitating the accomplishment of specified tasks and executing business processes as intended. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Performance Efficiency

Sub-characteristic **Time-behaviour** analyzes the response and processing times and throughput rates of a system, when performing its functions. Contextualizing for SOA, we can say response timeis related to time spent by services to process a request and return a response.

Sub-characteristic **Resource Utilization** addresses the amounts and types of resources used by a product or system, when performing its functions. Generally, software should make a best use of resources such as processor capacity, memory usage, disk capacity and network bandwidth. With re-gard to SOA, service based applications interact with other systems exchanging messages over the network.

Sub-characteristic **Capacity** means the degree to which the maximum limits of a product or system parameter meet requirements. This quality subcharacteristic was vaguely defined by ISO. It isdifficult to set the real meaning of this concept. We suppose that capacity can be the application ability to support several accesses at the same time without performance variation. In this sense, ca- pacity is related with availability of a service even with many access at the same time.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree

2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

10. **Time Behavior:** The service demonstrates time behavior by efficientlyprocessing requests and returning responses. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

11**. Resource Utilization:** The services utilize resources, such as servers, to access information from other applications. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

12. **Capacity:** The service exhibit capacity by remaining operational even with a large number of simultaneous accesses. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Compatibility

Sub-characteristic **Co-existence** is concerned with the degree to which a product can perform its re-quired functions efficiently while sharing a common environment and resources with other products,without detrimental impact on any other product.

Sub-characteristic **Interoperability** refers to the degree to which various modules within a system,such as image processing and AI model inference, can exchange and effectively utilize the informa-tion that has been shared, in addition to the service's capacity to handle diverse types of image for-mats from user inputs.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

13. Co-existence: The different composite services co-exist by sharing theuse of the same service operations. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

14. Interoperability: The service demonstrates interoperability by facilitating smooth interaction between different modules (image processing, AI model inference) and by supporting various types of image formats for user input. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Usability

Sub-characteristic **Appropriateness recognizability** refers to the degree to which users can recog-nize whether a system is appropriate for their needs.

Sub-characteristic **Learnability** is related to the degree to which a system can be used by specified users to achieve specified goals of learning to use the system in an easy way. From the SOA perspec-tive, this sub-characteristic refers to the degree to which a service can facilitate the understanding of its operation.

Sub-characteristic **Operability** is related to the degree to which a system exhibits operability througha user-friendly interface and comprehensive documentation that facilitate effective interaction and understanding of the service.

Sub-characteristic **User Error Protection** is related to the degree to which a system implements safeguards and mechanisms to prevent errors arising from inappropriate or unexpected user inputs.This involves incorporating input validation, data preprocessing steps, and providing clear user guidelines to mitigate the likelihood of user-induced errors and ensure a smoother user experience.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

15. Appropriateness recognizability: The users recognize the appropriateness of a service for their needs through the service description, including information on functionality and transmitted data types. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

16. Learnability: The service demonstrates learnability by facilitating the understanding of its operation. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

17. Operability: The service exhibits operability through a user-friendly interface and comprehensive documentation that facilitate effective interaction and understanding of the service. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

18. User Error Protection: The service provides user error protection by incorporating input validation mechanisms and data preprocessing steps that mitigate the likelihood of errors arising from inappropriate or unexpected user inputs. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Reliability

Sub-characteristic **Maturity** is related to the degree to which a system meets needs for reliability under normal operation. Clearly, maturity can be applied in SOA because whenever a service consumer requests some information it is expected that a response is returned.

Availability is a very important sub-characteristic in SOA applications. Availability consists in defining the degree to which a system, product or component is operational and accessible when re-quired for use. This concept can be associated with SOA by the fact that a service provider must beavailable when a service consumer requests some information.

Sub-characteristic **Fault Tolerance** refers to the degree to which a system operates as intended despite the presence of hardware or software faults. From the SOA perspective, fault tolerance can be applied in SOA applications as meaning that strategies must be performed when a failure happens on some hardware or software.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

19. Maturity: The service demonstrates maturity by consistently returning aresponse when a consumer requests information. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

20. Availability: The service is available when requested by users. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

21. Fault tolerance: Services can create strategies that may be performed when a failure happens on some hardware or software. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

22. Recoverability: The service can recover data when occurs some interruption or failure. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Security

Confidentiality is the degree to which a system ensures data are accessible only to those authorizedto have access. This quality sub-characteristic must be found in SOA applications because informa- tion shared by a service provider can be accessed only to an authorized service client.

Integrity refers to the degree to which a system, product or component prevents unauthorized access to, or modification of, software or data. Services must be developed to prevent unauthorizedaccess to, or modification of private data.

Sub-characteristic **Non-repudiation** is related to the degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later. In the SOA con- text, this concept can be the ability of a service provider to prove that information has been deliveredto a service consumer.

Sub-characteristic **Accountability** refers to the degree to which actions of an entity can be traceduniquely to the entity.

Authenticity refers to claim and identification of a subject or resource requests access to a certain information. A system built using a SOA approach may encompass services provided by thirdparty organizations. The identity of the external service provider should be authenticated.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

23. Confidentiality: The service maintains confidentiality by ensuring that shared information can only be accessed by authorized clients. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

24. Integrity: The service exhibit integrity by preventing unauthorized access to or modification of private data. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

25. Non-Repudiation: The service provider implements non-repudiation strategies to prove the delivery of information to a service consumer. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

26. Accountability: The services demonstrate accountability by being autonomous.

1	2	3	4	5

Strongly Disagree **Strongly Agree**

27. Authenticity: The service ensures authenticity by authenticating the identity of the external service provider. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Maintainability

Modularity is a quality sub-characteristic that defines the capacity of a system to be composed of components such that a change to one component has minimal impact on other components.

Reusability is an important quality sub-characteristic which indicates that an asset can be used in more than one system, or in building other assets. This definition can be extended to SOA. Service reusability is a SOA design principle that means services are reusable.

Sub-characteristic **Analyzability** indicates the degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modi- fied. Within SOA context, Analyzability can be an important quality attribute for services. In situations that a service need to be modified, analysis of which composite services use this service are neces- sary.

Modifiability is a quality sub-characteristic related to the capacity of software to be modified with- out introducing defects or degrading existing product quality. Modifiability increases as the indepen-dence between modules of the system increases because a change in one module can affect all modules that are dependent.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

28. Modularity: The service provider demonstrates modularity by hosting a network-accessible software module. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

29. Reusability: The services exhibit reusability by being reusable. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

30. Analyzability: The services be analyzed to understand the impact of changes when modifications are needed. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

31. Modifiability: The services demonstrate modifiability by being loosely coupled, reducing dependencies, and increasing adaptability. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

32. Testability: The services can be tested, for instance, through automated tools for functional testing. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Portability

Sub-characteristic **Adaptability** refers to the degree to which a product or system can effectively andefficiently be adapted for different or evolving hardware, software

Sub-characteristic **Installability** is related to the degree of effectiveness and efficiency with whicha product or system can be successfully installed and/or uninstalled in a specified environment.

Sub-characteristic **Replaceability** indicates the degree to which a product can replace another speci-fied software product for the same purpose in the same environment.

On a scale of 1 to 5, please rate the following statements based on your experience with the AI-as-a-Service system:

1 – Strongly Disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly Agree

33. Adaptability: The services demonstrate adaptability, considering the possibility of a platform change even though they run remotely on a server. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

34. Installability: The services exhibit installability with respect to potential platform changes while running remotely on a server. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

35. Replaceability: The services be replaced in the event of a platform change, given that they run remotely on a server. *

1	2	3	4	5

Strongly Disagree **Strongly Agree**

Comments and Suggestions

36. Additional feedback to improve the system and the study. *

37. Suggestions and Recommendations *

C. GANTT CHART

Week	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th	17th	18th	19th	20th	21st	22nd	23rd	24th	25th
Chapter 1																									
Introduction	■																			■					
Objectives	■																			■					
Significance of the Study	■																			■					
Scope and Delimitations	■																			■					
Validation from the Adviser			■																						
Chapter 2																									
Conceptual Framework		■																			■				
Recent State of Fisheries in the Philippines		■																							
FISHING METHODS AND TECHNIQUES IN THE PHILIPPINES				■																					
Impacts of Marine Pollution on The Philippine Fisheries		■																							
Precision Fishing and Aquaculture		■																							
Applications of Computer Vision in Fisheries			■																						
Synthetic Image Data Generation			■																						
Convolutional Neural Networks				■																					
Image Classification				■																					
INSTANCE SEGMENTATION AND OBJECT DETECTION				■																					
Validation from the Adviser				■						■	■	■							■	■	■				

Chapter 3																									
RESEARCH DESIGN																									
METHODS OF DATA GATHERING																									
THE DATASETS																									
ETHICAL CONSIDERATIONS																									
SOFTWARE DEVELOPMENT METHODS																									
REQUIREMENT SPECIFICATION																									
ARCHITECTURE AND DESIGN																									
UNIFIED MODELING LANGUAGE DIAGRAMS																									
COMPUTER VISION MODEL DESIGN																									
EVALUATION AND INFERENCE METRICS																									
DEPLOYMENT PLAN																									
Synthetc data generation																									
image_composition.py																									
coco_json_utils.py																									
Datasets																									
Custom Vision																									
Deep Fish																									
Marine Debris																									
Marine Garbage																									
Marine Animals																									
Market Marine Animals																									
Trash Annotations in Context (TACO)																									
Computer Vision Algorithms																									
Custom CNN																									
Mask R-CNN																									

YOLOv5																									
Detectron																									
Websites																									
Templates Draft																									
Image Data Generation																									
Aquaria																									
Documentation																									
Create Colab Account																									
Create GitHub Repository and Push Project																									
Dataset Feasibility/Training/Deployment																									
Custom Vision																									
Deep Fish																									
Marine Debris																									
Marine Garbage																									
Marine Animals																									
Market Marine Animals																									
Trash Annotations in Context (TACO)																									

D. RESEARCH EXPENDITURE

Expenditure Category	Amount (in ₱)
Custom Environment	
- Big Aquarium	2,500
- Small Aquarium	1,000
- Filter Wool	600
- Fish Capturing pay	500
- Fish food	300
GPU Google Colab Subscription	
- November 2022	580.41
- December 2022	1662.04
- January 2023	556.75
- February 2023	1639.62
- March 2023	548.32
- April 2023	542.42
- May 2023	2750.96
- June 2023	560.21
Miscellaneous Fee	
- Thesis panel fee	5,100
- Defense materials	679.75
Total Cost	18,720.48

Custom Environment (₱4,900): This section outlines the costs associated with setting up and maintaining the aquarium environments necessary for the research. It includes the costs of large and small aquariums, filter wool for water filtration, the fees for capturing fish, and fish food.

GPU Google Colab Subscription (₱8,040.73): These costs cover the subscription fees for using Google Colab's GPU for running computations necessary for the project. The subscription fees varied each month based on the prevailing exchange rate, and there were some additional charges in months where extra usage was required.

Miscellaneous Fee (₱5,779.75): This part of the breakdown encapsulates costs that do not fit into the above categories. It includes the fees paid to the thesis panel, as well as materials needed for the defense.

The total cost of the research project, combining all of the above, came out to ₱18,720.48. It's important to note that this represents the direct financial cost and doesn't include indirect costs such as time and effort expended by the research team.

E. USER GUIDE

Aquaria User Guide

Welcome to Aquaria, your comprehensive AI-powered marine image analysis tool. Accessible via the web, Aquaria leverages advanced computer vision models to analyze and provide insights on your marine images. Follow our user guide to navigate through the various services provided by Aquaria and start your marine image analysis journey today.

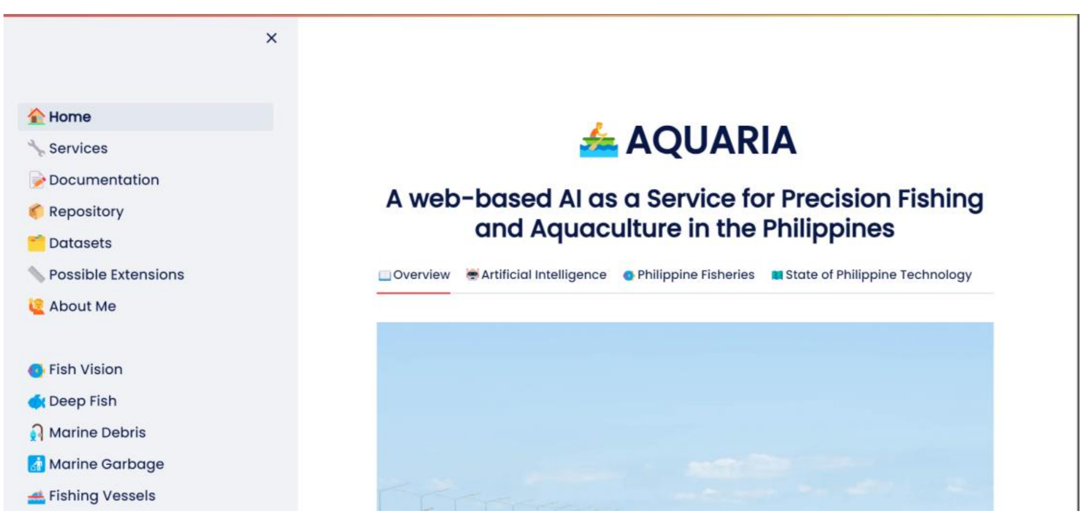
Aquaria can be accessed through the following URL: **https://aquaria-thesis.streamlit.app/**

1. Accessing Aquaria

Upon opening the Aquaria link, you will be prompted to enter a password. For this guide, the password is **'admin'**. Enter the password and proceed to the Aquaria homepage.

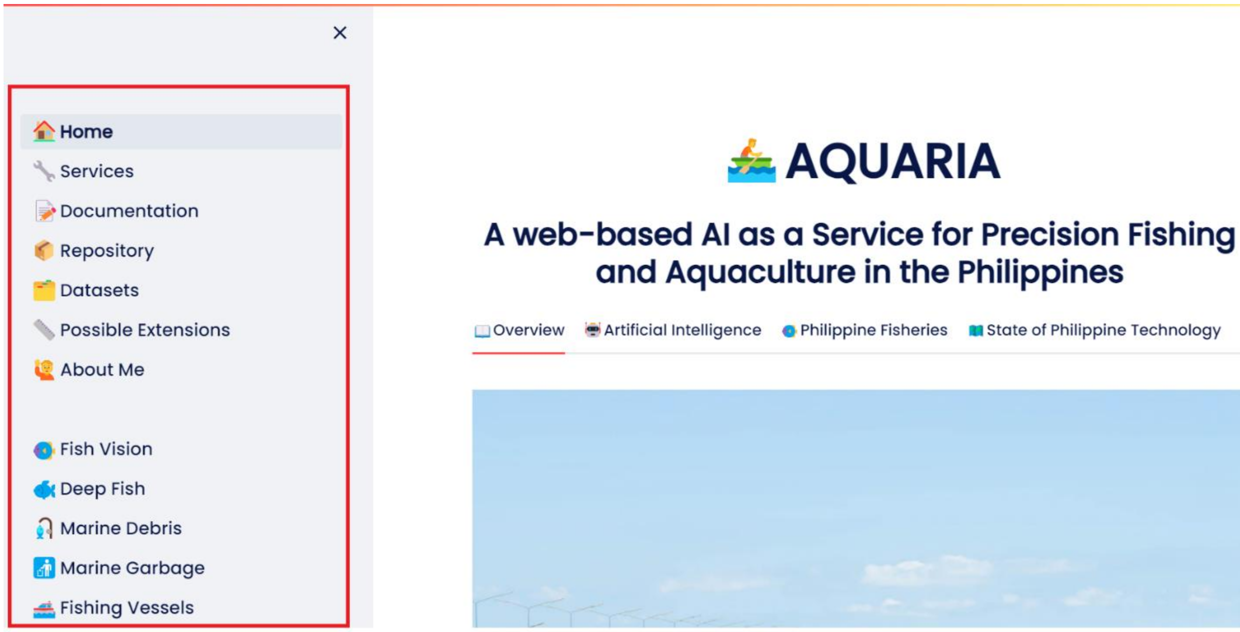
2. The Homepage

The Aquaria homepage serves as an overview page providing an outline on the state of fisheries, AI, and technology in the Philippines.



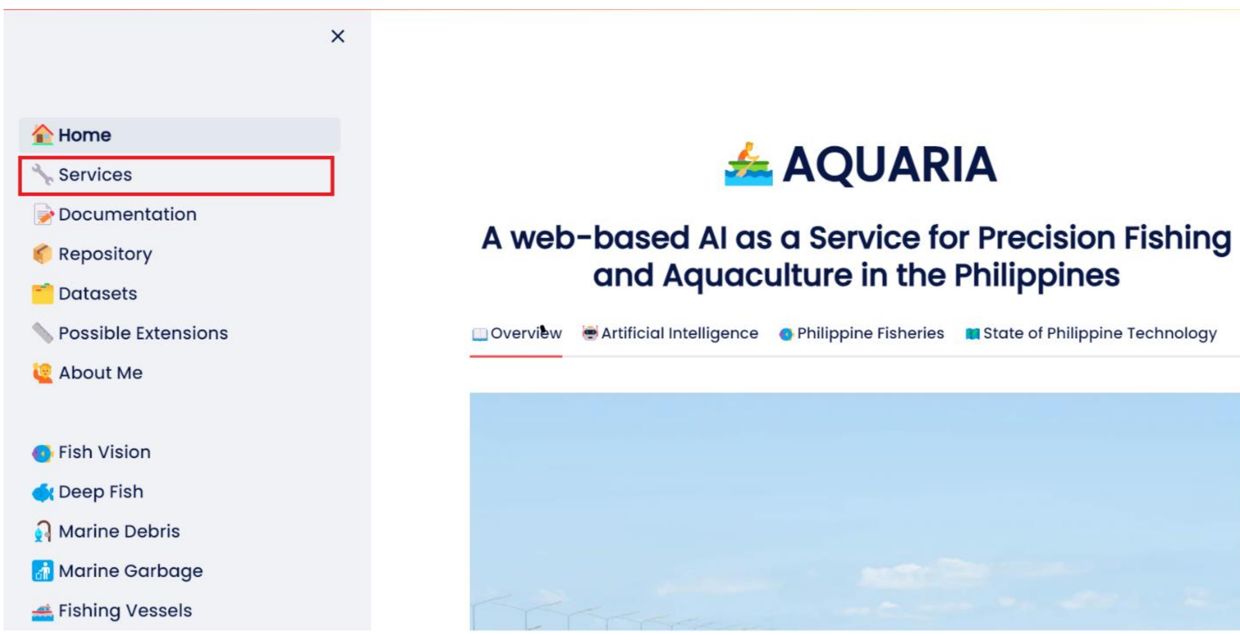
3. Navigation

On the left of the homepage, you will find a sidebar which contains various tabs that lead to the different services provided by Aquaria.



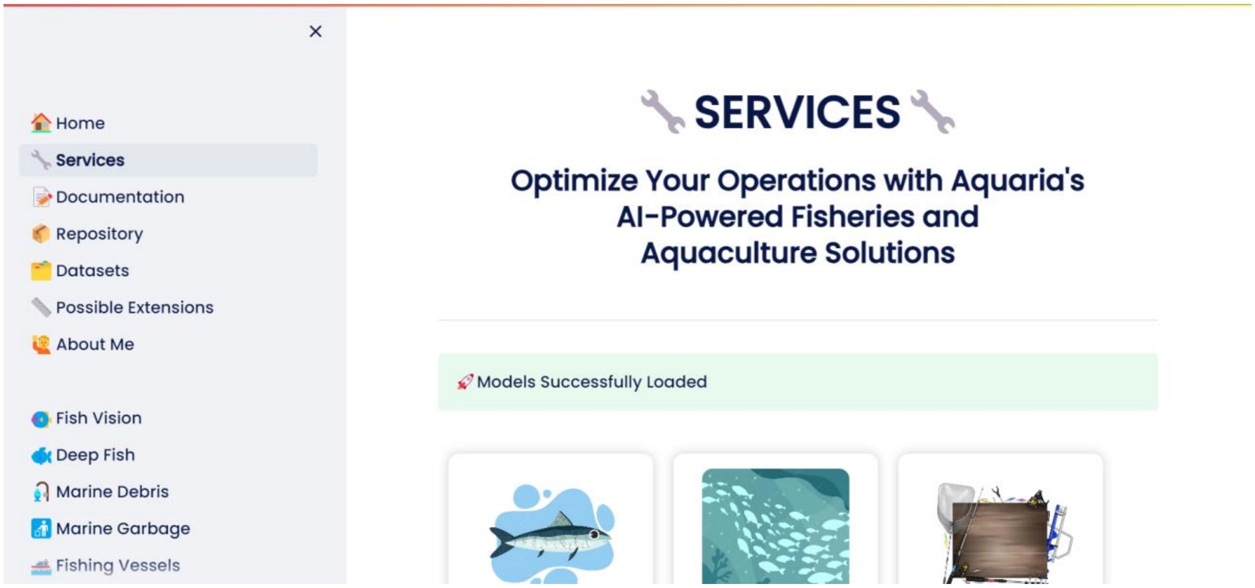
4. Accessing Services

To access the computer vision services, click on the 'Service' tab. If the model weights for the service you chose are not yet downloaded, Aquaria will automatically download these.



5. Model Download

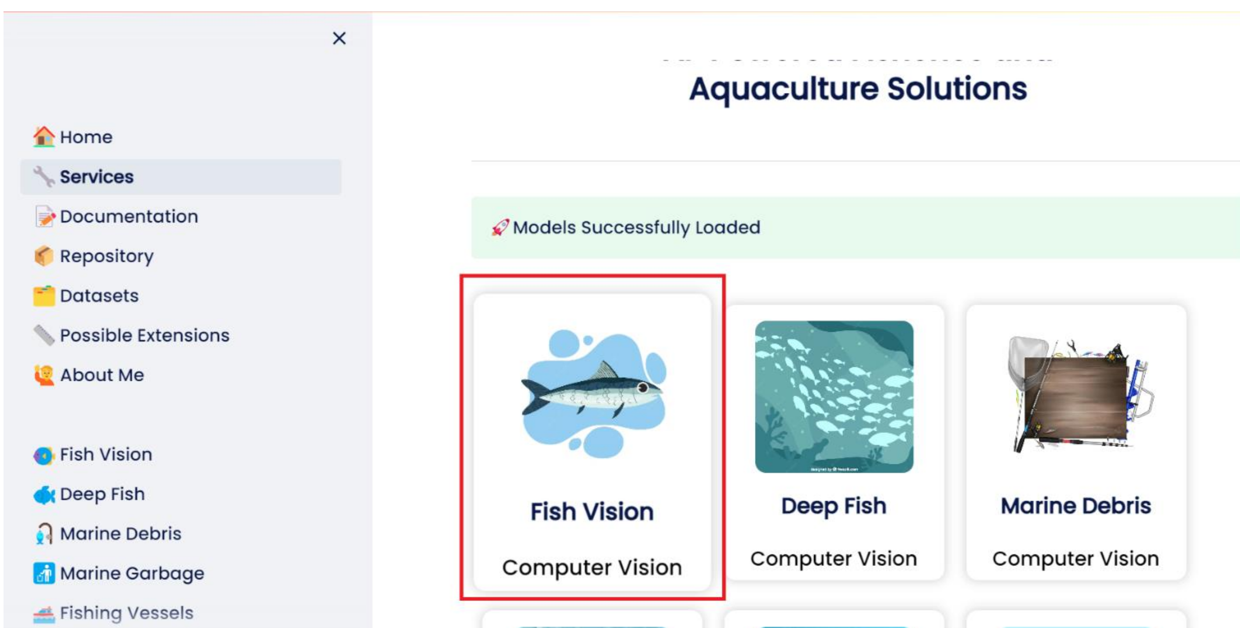
If the weights are already downloaded, you will see the message: "🚀 Models Successfully Loaded".

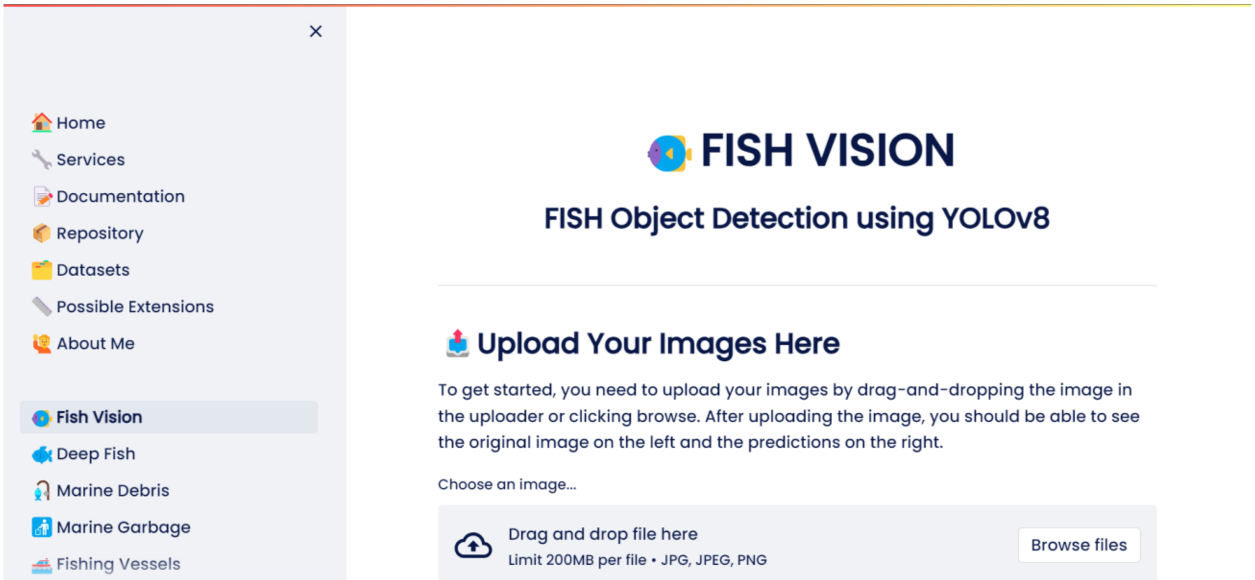


6. Using a Service

After the model weights have been downloaded, you can now access and use the services. Select the service you wish to use.

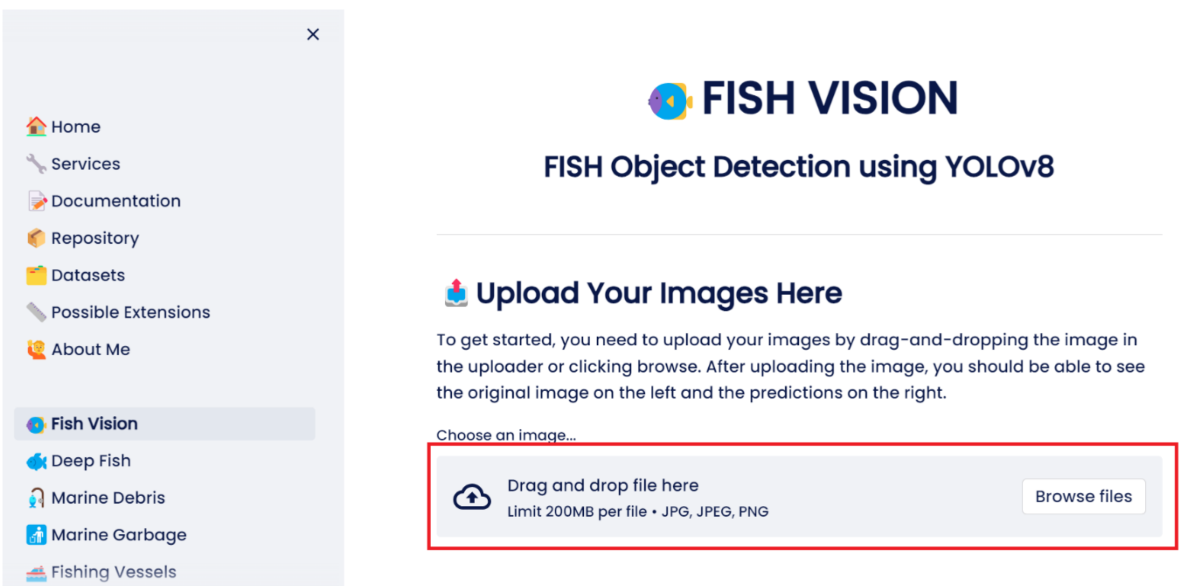
For instance, if you select 'Fish Vision Service', you will be directed to the Fish Vision Service page.





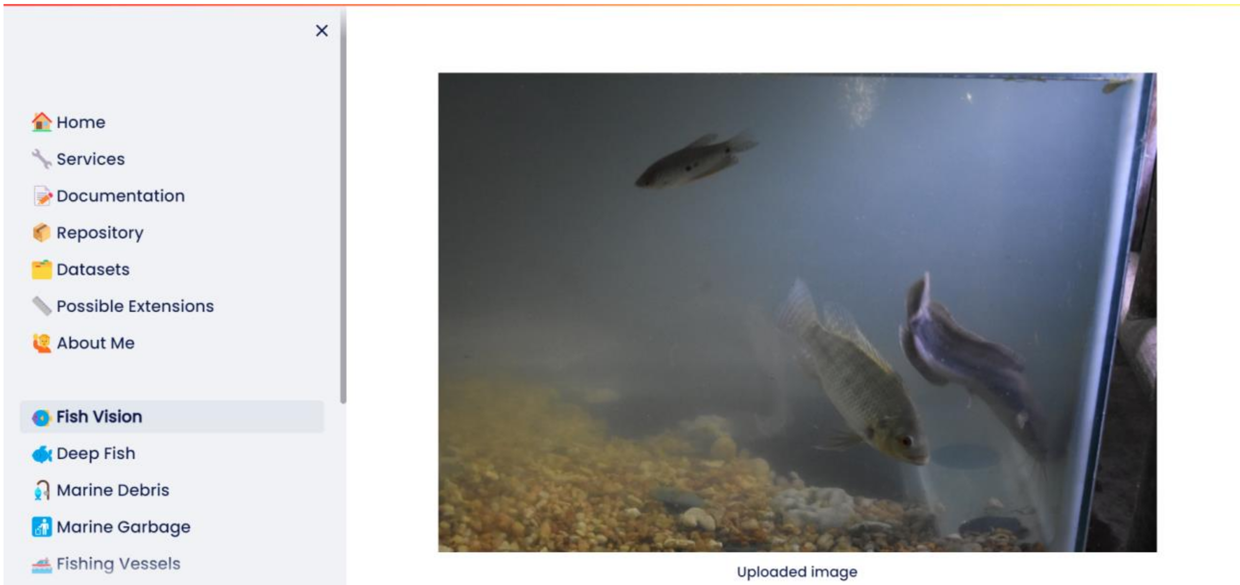
7. Uploading an Image

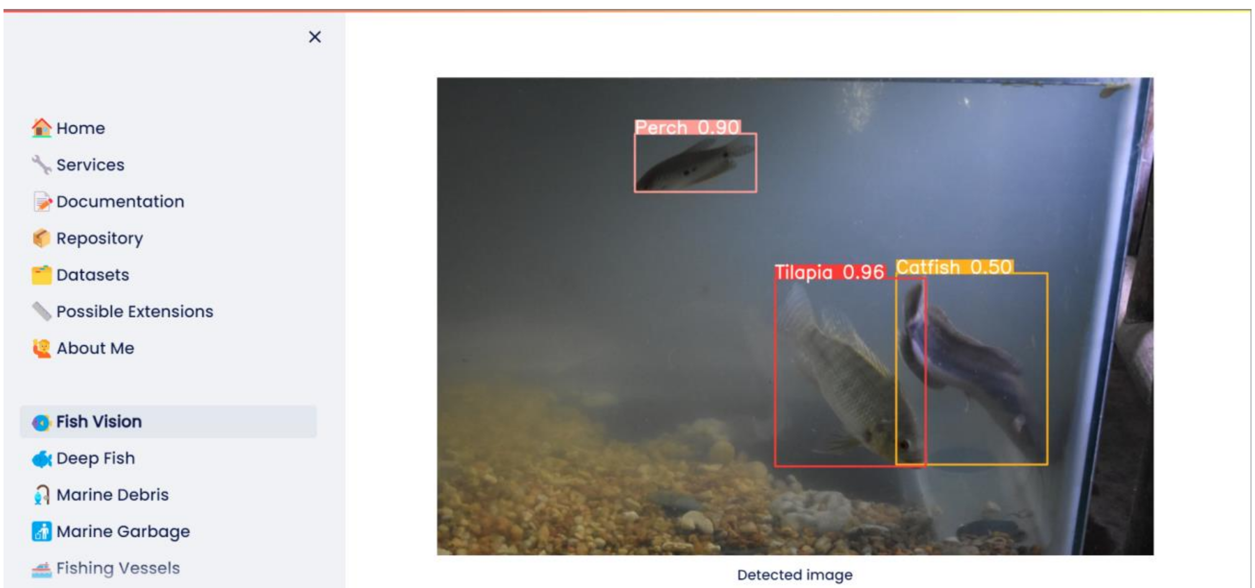
On the Fish Vision Service page, you will be prompted to upload a photo. Click on the 'Browse Files' button, select your photo, and confirm your selection to upload the image. You may also choose to drag-and-drop your photo.



8. Model Processing and Results Display

Once the image is successfully uploaded, the computer vision model processes the image in the background and subsequently displays the predicted output to the user.





You can now use Aquaria to analyze your marine images using our AI-powered computer vision models.